

2013

## Optimization of a Coded-Modulation System with Shaped Constellation

Xingyu Xiang  
*West Virginia University*

Follow this and additional works at: <https://researchrepository.wvu.edu/etd>

---

### Recommended Citation

Xiang, Xingyu, "Optimization of a Coded-Modulation System with Shaped Constellation" (2013). *Graduate Theses, Dissertations, and Problem Reports*. 207.  
<https://researchrepository.wvu.edu/etd/207>

This Dissertation is protected by copyright and/or related rights. It has been brought to you by the The Research Repository @ WVU with permission from the rights-holder(s). You are free to use this Dissertation in any way that is permitted by the copyright and related rights legislation that applies to your use. For other uses you must obtain permission from the rights-holder(s) directly, unless additional rights are indicated by a Creative Commons license in the record and/ or on the work itself. This Dissertation has been accepted for inclusion in WVU Graduate Theses, Dissertations, and Problem Reports collection by an authorized administrator of The Research Repository @ WVU. For more information, please contact [researchrepository@mail.wvu.edu](mailto:researchrepository@mail.wvu.edu).

# Optimization of a Coded-Modulation System with Shaped Constellation

by

Xingyu Xiang

Dissertation submitted to the  
College of Engineering and Mineral Resources  
at West Virginia University  
in partial fulfillment of the requirements  
for the degree of

Ph.D.  
in  
Electrical Engineering

Brian D. Woerner, Ph.D.  
Daryl S. Reynolds, Ph.D.  
John Goldwasser, Ph.D.  
Natalia A. Schmid, Ph.D.  
Matthew C. Valenti, Ph.D., Chair

Lane Department of Computer Science and Electrical Engineering

Morgantown, West Virginia  
2013

Keywords: Information Theory, Constellation Shaping, Capacity, LDPC code,  
Optimization, EXIT chart

Copyright 2013 Xingyu Xiang

## Abstract

### Optimization of a Coded-Modulation System with Shaped Constellation

by

Xingyu Xiang

Ph.D. in Electrical Engineering

West Virginia University

Matthew C. Valenti, Ph.D., Chair

Conventional communication systems transmit signals that are selected from a signal constellation with uniform probability. However, information-theoretic results suggest that performance may be improved by shaping the constellation such that lower-energy signals are selected more frequently than higher-energy signals. This dissertation presents an energy-efficient approach for shaping the constellations used by coded-modulation systems. The focus is on designing shaping techniques for systems that use a combination of amplitude-phase shift keying (APSK) and low-density parity check (LDPC) coding. Such a combination is typical of modern satellite communications, such as the system used by the DVB-S2 standard.

The system implementation requires that a subset of the bits at the output of the LDPC encoder are passed through a nonlinear shaping encoder whose output bits are more likely to be a zero than a one. The constellation is partitioned into a plurality of sub-constellations, each with a different average signal energy, and the shaping bits are used to select the sub-constellation. An iterative receiver exchanges soft information among the demodulator, LDPC decoder, and shaping decoder. Parameters associated with the modulation and shaping code are optimized with respect to information rate, while the design of the LDPC code is optimized for the shaped modulation with the assistance of extrinsic-information transfer (EXIT) charts. The rule for labeling the constellation with bits is optimized using a novel hybrid cost function and a binary switching algorithm.

Simulation results show that the combination of constellation shaping, LDPC code optimization, and optimized bit labeling can achieve a gain in excess of 1 dB in an additive white Gaussian noise (AWGN) channel at a rate of 3 bits/symbol compared with a system that adheres directly to the DVB-S2 standard.

# Acknowledgements

This dissertation could never be finished without the assistants and contributions from a great many people around. I would like to extend my appreciation especially to the following.

First and foremost, my deepest gratitude is to my dissertation committee chairman and advisor, Dr. Matthew C. Valenti, who is my first foreign advisor and also the best advisor I have ever met. I can still remember the exciting scene that I received the email from Dr. Valenti about his research plan together with the official invitation. His excellent guidance, patience, and enthusiasm attitude made me overcome the problems came across the whole graduate career. He provides the support in almost every possible way, I appreciate his time, idea, editing and funding in the completion of this and other documents. It is an honor to work with him and his instructions will continue benefit my future lives.

To the other members of my committee, Dr. Brian Woerner, Dr. Daryl Reynolds, Dr. Natalia Schmid and Dr. John Goldwasser, thank you for your feedback and suggestions to make this dissertation more thorough and rigorous.

I spend most of my professional and personal time in the wireless communication research lab, from where I obtain both friendship and illuminating advice. I am grateful for the colleagues who stayed with me in the past few years: Terry R. Ferret, Mohammad Fanaei, Salvatore Talarico, Aruna Sri and Chandana Nannapaneni. Particularly I would like to acknowledge Terry for helping me settle down on my first day of arrival, he also help Dr. Valenti to maintain the cluster system to provide us the extra computing power. Without all of their contributions to the grid computing, the results of this dissertation would take much more time to be finished. Other graduate students that I have had the pleasure to be alongside of are Jinyu Zuo, Ricky Hussmann, Yuan Li and Wentian Zhou.

My time at West Virginia University was enjoyable due to the beautiful environment and the unique graduate student experience, it becomes a part of my life and thanks to all the professors and friends to make it memorable. Special recognition goes out to my family I

would like to thank my parents for all their love and encouragement. They supported me and trusted me unconditionally even in the tough times. Lastly, I want to thank the WVU staff and my friends in China, America and elsewhere for their support and encouragement.

# Contents

<b>Acknowledgements</b>	<b>iii</b>
<b>List of Figures</b>	<b>vi</b>
<b>List of Tables</b>	<b>x</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Digital Communication System Model . . . . .	2
1.2 Modulation . . . . .	3
1.3 Channel Coding . . . . .	4
1.3.1 Convolutional Codes . . . . .	5
1.3.2 Turbo Code . . . . .	6
1.3.3 Linear Block Codes . . . . .	8
1.3.4 Low-Density Parity-Check (LDPC) Codes . . . . .	9
1.4 Satellite Communication . . . . .	15
1.4.1 DVB-S2 Standard . . . . .	16
1.4.2 APSK . . . . .	17
1.4.3 TWTA . . . . .	18
1.4.4 DVB-S2 Performance . . . . .	22
1.5 Thesis Outline . . . . .	24
<b>2 Channel Capacity and Constellation Shaping</b>	<b>26</b>
2.1 Channel Capacity . . . . .	26
2.1.1 The Unconstrained Capacity . . . . .	27
2.1.2 Capacity under Coded Modulation . . . . .	30
2.1.3 Capacity Calculation . . . . .	34
2.2 Constellation Shaping . . . . .	36
2.2.1 Shaping System Model . . . . .	38
2.2.2 Shaping Strategies . . . . .	41
2.2.3 Shaping for 16-APSK and 32-APSK . . . . .	41
2.3 Parameter Optimization . . . . .	44
2.4 PAPR Consideration . . . . .	49
2.5 Summary . . . . .	50

<b>3</b>	<b>Constellation Shaping Implementation</b>	<b>52</b>
3.1	Transmitter Structure . . . . .	53
3.1.1	Constellation Shaping with 16-PAM . . . . .	54
3.1.2	Constellation Shaping with 32-APSK . . . . .	54
3.2	Receiver Implementation . . . . .	56
3.2.1	The Demodulator . . . . .	58
3.2.2	The Shaping Decoder . . . . .	59
3.2.3	Receiver with Turbo Decoder . . . . .	60
3.2.4	Receiver with LDPC Decoder . . . . .	62
3.3	Numerical Results . . . . .	63
3.3.1	Results When using Turbo Codes . . . . .	64
3.3.2	Results when using LDPC Codes . . . . .	68
3.3.3	Receiver Complexity Consideration . . . . .	70
3.4	Conclusion . . . . .	72
<b>4</b>	<b>EXIT Chart and LDPC Code Optimization</b>	<b>73</b>
4.1	EXIT Charts Based Analysis . . . . .	73
4.1.1	Exit Charts for Uniform System . . . . .	73
4.1.2	Exit Charts for Shaped System . . . . .	81
4.2	LDPC Codes Optimization . . . . .	86
4.2.1	Codes Designed for AWGN Channel . . . . .	87
4.2.2	Codes Designed for Fading Channel . . . . .	93
4.3	Short Cycles in LDPC Codes . . . . .	96
4.4	Conclusion . . . . .	98
<b>5</b>	<b>Constellation Optimization</b>	<b>99</b>
5.1	Mapping Optimization Strategy . . . . .	100
5.1.1	Constellation Labeling Optimization . . . . .	105
5.1.2	Binary Switching Algorithm and the Cost Functions . . . . .	106
5.1.3	Mapping Optimization for Uniform 32-APSK . . . . .	111
5.1.4	Mapping Optimization for Shaped 32-APSK . . . . .	112
5.2	Numerical Results with Optimized LDPC Codes and Symbol Mapping . . .	115
5.3	Optimization of the Input Alphabet . . . . .	117
5.4	Conclusion . . . . .	122
	<b>References</b>	<b>124</b>

# List of Figures

1.1	Block diagram for a basic digital communication system. . . . .	2
1.2	Bit Error Rate of Uncoded Modulations . . . . .	5
1.3	Convolutional encoder . . . . .	6
1.4	Structure of turbo encoder . . . . .	7
1.5	Structure of turbo decoder . . . . .	8
1.6	Tanner Graph corresponding to $\mathbf{H}_1$ , the squares represent check nodes and the circles represent check nodes. . . . .	11
1.7	16-APSK . . . . .	18
1.8	Caption for LOF . . . . .	19
1.9	Distorted 16-QAM on the outputs of non-linear TWTA . . . . .	20
1.10	Distorted 32-APSK on the outputs of non-linear TWTA . . . . .	20
1.11	Magnitude Predistortion linearizers generate a response opposite to a TWTA's characteristics . . . . .	21
1.12	Phase Predistortion linearizers generate a response opposite to a TWTA's characteristics . . . . .	21
1.13	Predistorted QAM and 32-APSK on the outputs of non-linear TWTA . . . .	22
1.14	Performance of DVB-S2 standardized LDPC coded over AWGN channel . . .	23
1.15	Caption for LOF . . . . .	24
2.1	Unconstrained AWGN channel capacity and SIR with equiprobable M-QAM(PSK) inputs . . . . .	32
2.2	A BICM system structure which contains the CM system. The block 'Enc' is the binary channel encoder, 'Π' is a bit-level interleaver and 'Φ' is a memory-less mapper. The blocks at the receiver side are the reverse processes. . . . .	32
2.3	CM and BICM capacity for 16-QAM with uniform distribution, the BICM capacities are generated with two labeling rules . . . . .	34
2.4	Information rate results of uniform 32-APSK modulation generated by two methodologies . . . . .	37
2.5	Transmitter structure. . . . .	38
2.6	16-PAM constellation. . . . .	41
2.7	16-APSK constellation. . . . .	42
2.8	32-APSK constellation. . . . .	43



2.9	Information rate (in bits per channel use) of 16-APSK, 16-PAM and 32-APSK with uniform input distributions over AWGN and ergodic Rayleigh fading channels. The rates for APSK modulation are maximized over the permissible ring radius ratios. . . . .	45
2.10	Information rate of nonuniform APSK and 16-PAM. The number of shaping bit used is indicated. The solid black line on the right of each subfigure corresponds to a uniform input distribution. The results of AWGN Channel are shown in the first row, the results of ergodic Rayleigh fading channel are shown in the second row. Subfigures in leftmost column: 16-APSK with 1 or 2 shaping bits. Subfigures in the middle column: 16-PAM with 1 or 2 shaping bits. Subfigures in last column: 32-APSK with 1, 2, or 3 shaping bits. . . . .	47
2.11	Shaping gain of 16-PAM, 16-APSK and 32-APSK over AWGN and ergodic Rayleigh fading channel. For 16-PAM and 32-APSK, 1 shaping bits are used. For 16-APSK, 2 shaping bits are used . . . . .	49
2.12	The value of $p_0$ that maximizes the information rate for nonuniform APSK in AWGN and ergodic Rayleigh fading. For 16-APSK, 2 shaping bits are used, while for 16-PAM and 32-APSK, 1 shaping bit is used. . . . .	50
2.13	Peak-to-average power ratio (PAPR) of nonuniform 16-PAM, 32-APSK and 16-APSK as a function of $p_0$ and the number of shaping bits. For 16-APSK, $\gamma = 2.57$ , and for 32-APSK, $\gamma = \{2.64, 4.64\}$ . . . . .	51
3.1	32-APSK constellation. . . . .	55
3.2	Receiver structure. . . . .	56
3.3	Turbo decoder . . . . .	61
3.4	LDPC decoder . . . . .	62
3.5	Bit-error rate of 16-PAM in AWGN and fading channel . . . . .	66
3.6	Bit-error rate of 32-APSK in AWGN and fading channel . . . . .	67
3.7	Bit-error rate of LDPC-coded 16-PAM at rate $R = 3$ bits/symbol both with and without shaping. Curves are shown for AWGN and Rayleigh Fading channel. . . . .	68
3.8	Bit-error rate of LDPC-coded 32-APSK in AWGN at rate $R = 3$ bits/symbol both with and without shaping. Curves are shown for the unshaped (uniform) system using BICM and BICM-ID. Curves are shown for three shaping codes, and the shaped system uses BICM-ID. . . . .	69
3.9	Bit-error rate for the same system described in Fig. 3.2 in fully-interleaved Rayleigh fading. . . . .	70
3.10	Average number of BICM-ID global iterations required for correcting all codeword errors. . . . .	71
4.1	Demodulator mutual information transfer characteristics for 16-QAM with 'SP' labeling . . . . .	75
4.2	Demodulator mutual information transfer characteristics for different 16-QAM mappings at $\mathcal{E}_s/N_0 = 4.5\text{dB}$ . . . . .	76
4.3	Decoder mutual information transfer characteristics for rate 1/2 convolutional code with different code memory . . . . .	77

4.4	EXIT chart for 16-QAM modulation with 'SP' mapping at $\mathcal{E}_s/N_0 = 6\text{dB}$ , averaged decoding trajectories are also shown . . . . .	78
4.5	VND curves for LDPC codes over AWGN channel . . . . .	80
4.6	CND EXIT curves . . . . .	80
4.7	Exit chart for the uniform system with DVB-S2 standard LDPC code and 32-APSK modulation at $\mathcal{E}_b/N_0 = 4.93\text{ dB}$ . . . . .	82
4.8	EXIT chart for BICM-ID using M=32 APSK and overall system rate R=3.85 over AWGN channel at $\mathcal{E}_b/N_0 = 7\text{dB}$ . Different decoder characteristic curves correspond various channel decoder iterations. From left to right, the six decoder curves correspond to 1 through 6 iterations. . . . .	84
4.9	EXIT chart for the shaped system described in chapter 3 using M=32 APSK and overall system rate R=3 over AWGN channel at $\mathcal{E}_b/N_0 = 4.76\text{dB}$ . . . . .	85
4.10	Curve fit for an LDPC code at code rate $R = 3/5$ with uniform 32-APSK constellation, $E_b/N_0 = 4.5\text{dB}$ . . . . .	88
4.11	Bit-error rate of 32-APSK in AWGN at rate $R = 3$ bits/symbol with a length $n_c = 64800$ LDPC code. The solid lines are for the uniform system; the rightmost curve shows the performance of uniform modulation with a standard DVB-S2 code, while the two curve to its left show the performance of uniform modulation with optimized $D = 3$ and $D = 4$ LDPC codes, respectively. The dashed lines are for the shaped system; with the left two curves show the performance of shaped modulation with optimized $D = 3$ and $D = 4$ LDPC codes, and the third curve from left shows the performance of shaped modulation with a standard DVB-S2 code . . . . .	89
4.12	EXIT chart for BICM-ID using M=32 APSK and overall system rate R=3 over AWGN channel at $\mathcal{E}_b/N_0 = 4.73\text{dB}$ . . . . .	91
4.13	Bit-error rate of 32-APSK in AWGN at rate $R = 3$ bits/symbol with short length LDPC codes ( $n_c = 16200$ ). The solid lines are for the uniform system; from right to left, the curves show uniform system using $R_c = 3/5$ DVB-S2 standard code, uniform system with optimized code with $D = 3$ and $D = 4$ . The dashed lines are for the shaped system; from left to right, the curves show performance using optimized $R_c = 9/14$ LDPC code with $D = 4$ and $D = 3$ combined with (3, 2) shaping code, the third curve from left shows the performance of standard $R_c = 2/3$ code with (4, 2) shaping code . . . . .	92
4.14	Bit-error rate of 32-APSK in ergodic Rayleigh fading channel at rate $R = 3$ bits/symbol with a length $n_c = 64800$ LDPC code. The three curves on the right side are for the uniform system, the three curves on the left side are for the shaped systems. . . . .	94
4.15	Bit-error rate of 32-APSK in ergodic Rayleigh fading channel at rate $R = 3$ bits/symbol with a length $n_c = 64800$ LDPC code. The solid lines are for the uniform system, The dashed lines are for the shaped system. . . . .	95
4.16	The form of a 4 cycle in a parity-check matrix . . . . .	96
4.17	Bit-error rate of shaped 32-APSK at rate $R = 3$ bits/symbol with a length $n_c = 64800$ LDPC code over AWGN channel. The solid lines are results with the LDPC codes without 4 cycles, the dashed lines are for the LDPC codes with 4 cycles. . . . .	97

5.1	Four mappings of 8-PSK constellation . . . . .	101
5.2	16-APSK with mapping defined in DVB-S2 standard. Decision regions for bit $m = 1, 2, 3, 4$ with no a priori information. Shaded region is the decision region where bit $m$ is 0 . . . . .	102
5.3	The 16-APSK symbol with label (0000) is transmitted, decision regions for the first bit equals to 0 when 1, 2, 3 bits are known, respectively. The labeling is from DVB-S2 standard. . . . .	103
5.4	Bit-error rate of 8-PSK in AWGN at rate $R = 2$ bits/symbol with a length $n_c = 64800$ DVB-S2 LDPC code over AWGN channel. The dashed lines are for the uniform BICM systems; the solid lines are for the BICM-ID uniform system. . . . .	105
5.5	Bit-wise mutual information of 8-PSK with different mapping rules . . . . .	110
5.6	Bit-wise mutual information of 8-PSK with different mapping rules . . . . .	112
5.7	BER of 32-APSK in AWGN at rate $R = 3$ bits/symbol with a length $n_c = 64800$ DVB-S2 LDPC code. The dashed lines are the results with optimized mapping rule; the solid lines are for the system with DVB-S2 standard mapping rule. . . . .	113
5.8	Optimized Mapping of 32-APSK for shaped system . . . . .	114
5.9	Bit-error rate of shaped 32-APSK in AWGN at rate $R = 3$ bits/symbol. The LDPC code has rate $2/3$ and length $n_c = 64800$ LDPC codes. The (4, 2) shaping code is used. . . . .	114
5.10	Demodulator mutual information transfer characteristic for uniform 32-APSK at $\mathcal{E}_b/N_0 = 5$ dB . . . . .	116
5.11	Bit-error rate of 32-APSK in AWGN channel at rate $R = 3$ bits/symbol. The length of LDPC code is kept to be $n_c = 64800$ . . . . .	118
5.12	Gradient-search algorithm derived $M = 32$ signal constellation . . . . .	120
5.13	Bit-error rate of 32-APSK in AWGN at rate $R = 3$ bits/symbol with a length $n_c = 64800$ DVB-S2 LDPC code over AWGN channel. The dashed lines are for the BICM systems; the solid lines are for the BICM-ID system. . . . .	121
5.14	Gray 32-APSK with 2 rings . . . . .	122

# List of Tables

2.1	Minimum required $\mathcal{E}_b/N_0$ in AWGN for 16-PAM and M-APSK with $g$ shaping bits. The optimizing $p_0$ and $\gamma$ are shown. . . . .	48
2.2	Minimum required $\mathcal{E}_b/N_0$ in ergodic Rayleigh fading for 16-PAM and M-APSK with $g$ shaping bits. The optimizing $p_0$ and $\gamma$ are shown. . . . .	48
3.1	Parameters used for the systems with turbo code . . . . .	65
5.1	Values of Cost Functions related to Euclidean distance at $\mathcal{E}_s/N_0 = 8$ dB. . .	108

# Chapter 1

## Introduction

Digital communication is responsible for the transmission of useful data to one or more destinations through physical media or channels. Channel capacity is the highest information rate that can be transmitted reliably over a communication channel with arbitrarily small error probability [1]. The capacity value is given by the maximum of the mutual information between the input and output signals of the channel, where the maximization is with respect to the input distribution. The difference between the reliable system transmission rate and channel capacity becomes an important criteria to benchmark practical systems. Modern channel coding, e.g. turbo codes [2] and Low Density Parity Check (LDPC) codes [3, 4] approach channel capacity closely. When combined with other digital communication strategies, the system could perform even better. The work described in the following chapters is about certain optimization strategies to improve the performance of digital communications.

This chapter begins with an introduction to the basic components of digital communication systems. Channel coding, including turbo and LDPC codes, are briefly reviewed since they are a critical component of the systems implemented later in this dissertation. Next, satellite communication systems, including the Digital Video Broadcasting - Second Generation (DVB-S2) standard, are discussed with an emphasis on their performances. The last part is an overview of the reminder of the dissertation.

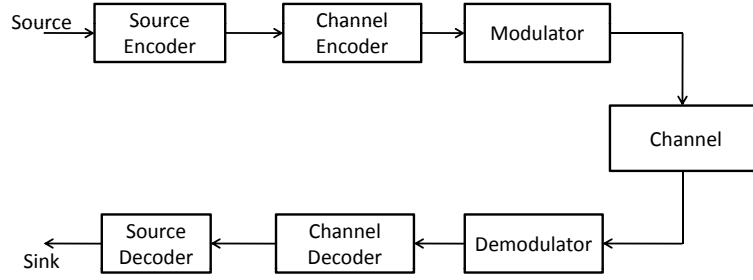


Figure 1.1: Block diagram for a basic digital communication system.

## 1.1 Digital Communication System Model

An elementary block diagram of a digital communication system is shown in Fig. 1.1. The aim of the communication system is for the sink to recover the messages from the source with certain reliability. The source is generally considered as a sequence of bits drawn according to a probabilistic model. The *source encoder* converts the information source sequence into an alternative sequence which is a more efficient representation. It compresses the information sequence into a shorter sequence using certain algorithms or coding strategies. Depending on the requirements of the application, the compression can be either lossless or lossy. The output of source encoder is sent to the *channel encoder*, which adds controlled redundancy that can be used by the receiver to detect and correct errors caused by the channel.

The *modulator* maps the output of channel encoder onto analog waveforms chosen from a finite set according to mapping and labeling rules. The modulated waveforms are then transmitted through channels that are corrupted by distortion and noise. Physical mediums like the atmosphere, wire lines, optical fibre, and other mediums could be used as the communication channel. The *demodulator* collects the distorted signals and computes an estimate of the probabilities of the modulated symbols. The output of the demodulator is sent to the *channel decoder*, which reduces the number of errors induced by the channel with the assistance of the redundancy introduced by the channel encoder. The *source decoder* decompresses and reconstructs the message, which is delivered to the sink. The difference or functions of the difference between the original signal and the reconstructed signal is a measure of the distortion or the demodulator-detector performance.

An actual digital communication system is far more sophisticated than the generalized description above. Some systems contain an amplifier in the transmitter or a quantizer at the input to the receiver. In addition, in some systems the demodulator and channel decoder work jointly or iteratively to recover the message.

## 1.2 Modulation

Analog modulation and digital modulation are the two basic types of modulation. Analog modulation uses analog information to modulate a carrier by using it to change the amplitude, frequency, phase, or some combination of the three. Demodulators for analog modulation are simple and cheap to implement, but analog modulation requires lots of power and is vulnerable to noise and distortion. The move from analog modulation to digital modulation provides more information capacity, compatibility with digital data services, higher data security and better quality communications. In digital modulation systems, an analog carrier is modulated by a discrete signal and the modulation can be considered as digital-to-analog conversion; the corresponding demodulation or detection is responsible for the opposite analog-to-digital conversion. For a wireless communication channel, the input bit stream must be represented by a high-frequency signal to facilitate transmission with an antenna of reasonable size.

Most communication systems belong to one of three categories: bandwidth efficient, power efficient, or cost efficient. Bandwidth efficiency or spectral efficiency refers to the ability to transmit at a high information rate over a limited bandwidth. Power efficiency describes the ability of the system to reliably send information at the lowest practical power level. The parameters to be optimized depend on the particular system. For digital terrestrial microwave radios, the bandwidth efficiency might be the highest priority factor, but power efficiency may be more important for the hand-held cellular phones since they need to run on a battery and for satellite communication systems, which must transmit over extremely long distances.

A simple example of the modulator is to map the binary digit 0 onto signal  $s_0(t)$  and the binary digit 1 onto signal  $s_1(t)$ . This is called *binary modulation*, and binary phase-

shift keying (BPSK) is one example of such modulation when the phases of  $s_0(t)$  and  $s_1(t)$  are separated by  $180^\circ$ . More generally, information can be transmitted by signals with  $M$  different phases. Such modulation by changing the phase of the carrier signal is called  $M$ -ary phase-shift keying (PSK). BPSK takes the highest level of noise or distortion of all the PSKs to make the demodulator reach an incorrect decision, but it can only modulate 1 bit per symbol and is unsuitable for high data-rate applications. One application of BPSK is for deep space telemetry. Modulation schemes with higher modulation order can convey more bits with one signal. In particular  $\log_2(M)$  bits can be grouped and mapped to one of  $M$  distinct signals, where  $M$  is the modulation order. However, as the order  $M$  of the modulation gets larger, the demodulator is more likely to mistake the actual transmitted signal for another one. The reason is that the  $M$  signals are placed relatively closer with higher values of  $M$ , and thus the channel noise can easily mislead the recovery of demodulator. Phase shift keying (PSK), pulse amplitude modulation (PAM), and quadrature amplitude modulation (QAM) are the most common modulation schemes. PSK is used extensively in applications including CDMA (Code Division Multiple Access) cellular service, PAM is used in some versions of the Ethernet communication standard, and QAM is used in applications including DVB-C (Digital Video Broadcasting - Cable) and 4G LTE (long-term evolution). Amplitude and Phase-shift keying (APSK), which is used in DVB-S2 (Digital Video Broadcasting - Second Generation), is an important constellation discussed in this dissertation.

The error performance of several modulation schemes over *additive white noise Gaussian* (AWGN) channel are shown in Fig. 1.2. There is no channel coding involved and the modulated symbols are passed directly through the noisy channel. The results illustrate the trend that the higher order modulation causes more errors although it can transmit more bits per symbol.

### 1.3 Channel Coding

The role of the channel code is to protect the bits to be transmitted over the channel from the corruption caused by noise, distortion, and interference. The channel encoder converts the input bits into an alternate sequence with redundancy, which provides immunity from



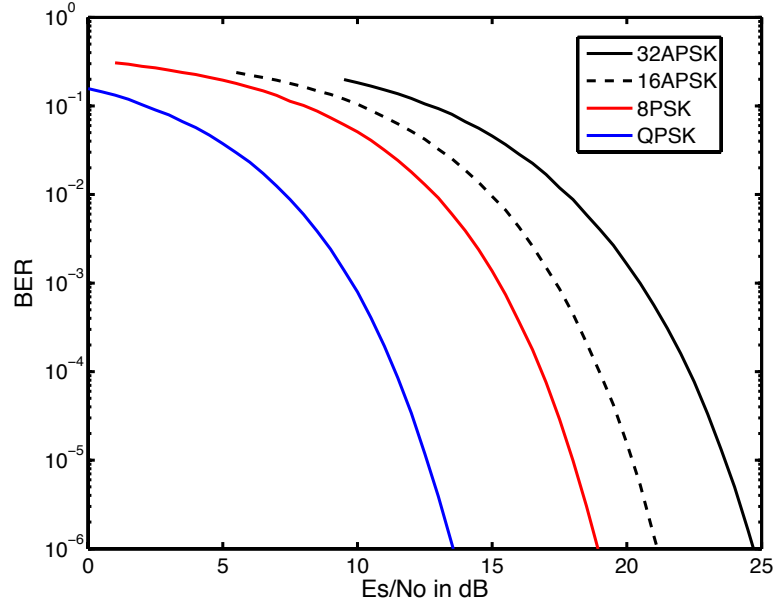
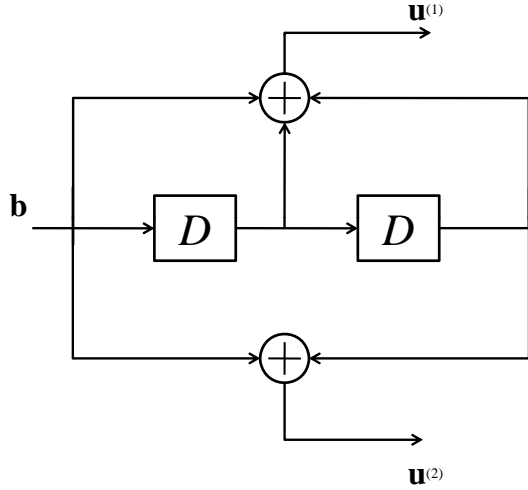


Figure 1.2: Bit Error Rate of Uncoded Modulations

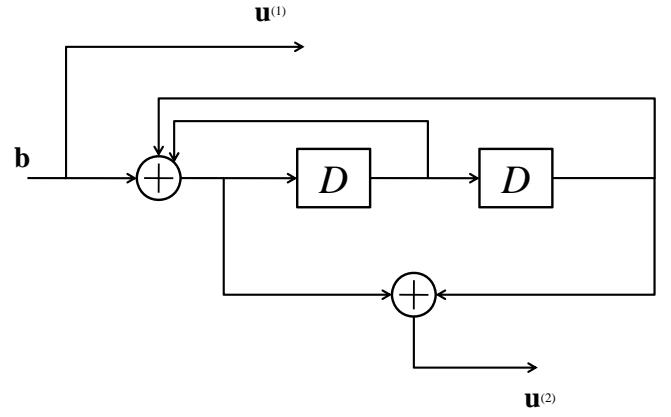
various channel distortions. The *code rate*  $R$  of a channel code is the ratio of the number of bits at the input of the channel encoder to the number of its output bits. The rate will always be less than unity because of the redundancy.

### 1.3.1 Convolutional Codes

Convolutional codes were first introduced by Elias in 1955[5]. The details about convolutional codes can be found in books by Johannessson and Zigangirov [6] and by Lin and Costello [7]. Convolutional codes are encoded by continuously passing the data bits through a finite state shift register. An example convolutional code is the four-state, rate  $1/2$  code whose encoder is depicted in Fig. 1.3(a). Notice that two code bits are produced for each data bit that is input into the encoder; i.e., the rate is  $1/2$ . The *state* is defined by the contents of the two binary memory cells, and this example, the encoder has four states. The convolutional codes can be expressed by  $(n, k, K)$  when the encoder reads  $k$  bits and outputs  $n$  bits at a time.  $K$  is the the number of memory cells or registers plus one, which is also called the *constraint length*. For binary codes, all operation are over the binary field and the top and bottom parts of the encoder act as two discrete-time finite-impulse-response filters. The top impulse response is  $\mathbf{g}^1 = [1 \ 1 \ 1]$  and the bottom impulse response is  $\mathbf{g}^{(1)} = [1 \ 0 \ 1]$ . The



(a) A four-state, rate-1/2 convolutional code encoder



(b) Recursive convolutional encoder

Figure 1.3: Convolutional encoder

output  $\mathbf{u}^{(1)}$  is the convolution of the input  $\mathbf{b}$  and  $\mathbf{g}^{(1)}$ , and similarly for the output  $\mathbf{u}^{(2)}$ . The convolution in time domain corresponds to a multiplication in the transform domain, where  $\mathbf{g}^{(1)}$  and  $\mathbf{g}^{(2)}$  can be used as the polynomial coefficients. The polynomials  $[1 + D + D^2, 1 + D^2]$  are called the *generator polynomial*, where  $D$  is the discrete-time delay operator generally used in the coding theory literature.

Two kinds of convolutional decoders are widely used. The first is the *maximum-likelihood sequence decoder* (MLSD), which is implemented via the Viterbi algorithm [8]. It minimizes the probability of codeword-sequence errors. The other one is the bit-wise *maximum a posteriori* (MAP) decoder which is implemented via the BCJR algorithm [9]. The MAP decoder minimizes the probability of information-bit errors.

### 1.3.2 Turbo Code

Turbo codes were first introduced in 1993 and became one of the most important breakthroughs in the history of coding theory [2]. A turbo encoder is shown in Fig. 1.4, which depicts a concatenation of two convolutional encoders separated by an interleaver. The uncoded data bits are encoded by the upper convolutional encoder, while the same uncoded bits

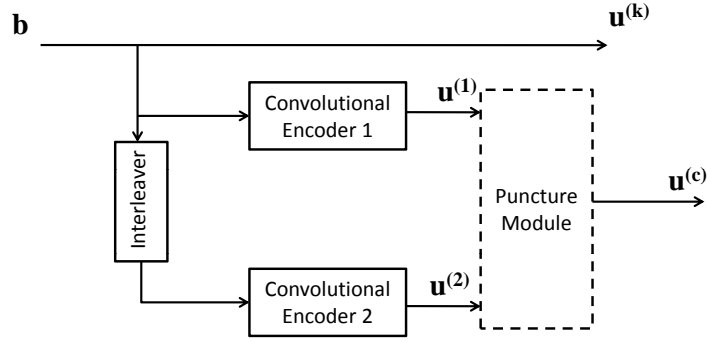


Figure 1.4: Structure of turbo encoder

are bit-wise interleaved and encoded by the lower convolutional encoder, which can be either identical or different with the upper one. A puncturing mechanism may exist to increase the rate of turbo code by deleting the output bits of one or both convolutional encoder.

The constituent convolutional codes of the original turbo code are known as *recursive systematic convolutional* (RSC) codes [2], which have a feedback loop to update the outputs. An example of the encoder of RSC code is shown in Fig. 1.3(b). The upper output in Fig. 1.3(a) is fed back to the initial stage of the shift register and the input bits appear directly in the output. The notation *systematic* means part of the output of the encoder is the input itself. The turbo code encoder contains two RSC encoders, and only one copy of the systematic output is transmitted while the other copy is discarded due to the redundancy.

A turbo decoder is presented in Fig. 1.5. The two decoders are soft-in/soft out (SISO) decoders, which are matched to the encoders in Fig. 1.4 [10]. The left SISO decoder (D1) receives noisy data and parity bits from the channel, corresponding to the bits  $\mathbf{u}^{(k)}$  and  $\mathbf{u}^{(1)}$  generated by the upper convolutional encoder E1, performs the BCJR algorithm, and produces the bit-wise log likelihood ratios (LLRs) for the data bits and parity bits. The extrinsic information on the systematic bits  $E_1 = U_1 - A_1 - Z_1$  is passed through the interleaver to become the a priori input  $A_2$  of the second decoder D2. The second decoder takes the permuted channel observation  $Z_2$  on the systematic bits  $\mathbf{u}^{(k)}$  and respective parity bits  $\mathbf{u}^{(1)}$  and feeds back extrinsic information  $E_2$  which becomes the a priori information  $A_1$  of the first decoder. Extrinsic information is set to be zero at the first decoder iteration, and

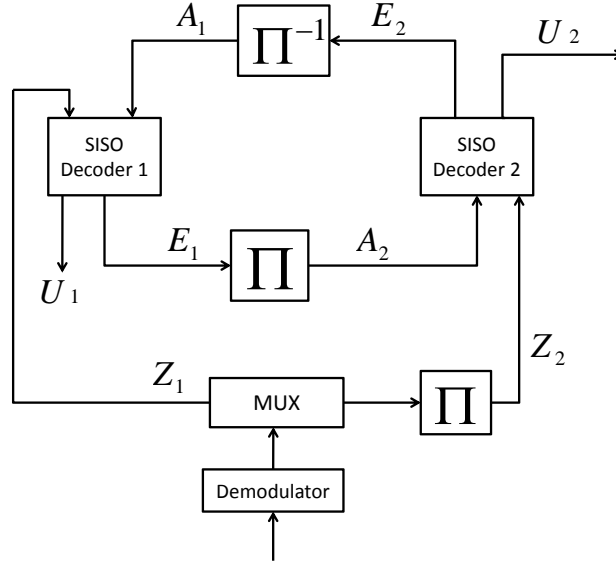


Figure 1.5: Structure of turbo decoder

it is updated during each iteration. The two decoders work in an iterative manner until they meet a stopping criteria.

### 1.3.3 Linear Block Codes

Assume that the transmitted message, or information sequence is a bit stream from the binary field  $\text{GF}(2)$ , i.e., only contains binary digits 0 and 1. Assume that the binary information sequence is divided into message blocks of fixed length  $k$ . The encoding process can be simply described by appending redundant bits to the current information bits. Then, the input message  $\mathbf{b}$  is transformed into a binary bit sequence  $\mathbf{u}$  with length  $n$ , where  $n > k$ ,  $\mathbf{u}$  is referred to as the *codeword* of the message  $\mathbf{b}$ . The bits appended to each message block have length  $n - k$ ; they contain no new information and just provide the capability of detecting and correcting transmission errors caused by the channel. Generating redundant bits with good error-correcting capability is a major concern in designing the channel codes.

There may exist  $k$  linearly independent codewords,  $\mathbf{g}_0, \mathbf{g}_1, \dots, \mathbf{g}_{k-1}$  such that every codeword  $\mathbf{u}$  is a linear combination of these  $k$  linearly independent codewords. If every codeword can be described as a linear combination of the  $k$  independent codewords, then it is a *linear code* denoted  $\mathcal{B}$ . The  $k$  independent codewords are the *basis* of the linear code.

Let each of these linearly independent codewords represent one row of a matrix. The size  $(k \times n)$  matrix is called the *generator matrix*  $\mathbf{G}$  and the encoding process can be expressed as the matrix multiplication of message  $\mathbf{b}$  and  $\mathbf{G}$ ; i.e.,  $\mathbf{u} = \mathbf{b} \cdot \mathbf{G}$ . An  $(n, k)$  linear block code has more than one such basis and the generator matrix is not unique. The *systematic* form of a generator matrix is  $\mathbf{G} = (\mathbf{I}_k, \mathbf{P})$ , where  $\mathbf{I}_k$  is a  $(k \times k)$  identity matrix and  $\mathbf{P}$  is a  $k \times (n - k)$  matrix. The linear block code generated by the generator matrix with this form is referred to as a linear systematic code, which can be divided into two parts, the  $k$  unchanged information bits and the  $(n - k)$  parity-check bits. This can be also shown by the expression  $\mathbf{u} = \mathbf{b}\mathbf{G}$  with the form  $(\mathbf{b}, \mathbf{b}\mathbf{P})$ , the first  $k$  components of  $\mathbf{u}$  are the same as the information sequence  $\mathbf{b}$ .

The dual code  $\mathcal{B}^\perp$  of linear code  $\mathcal{B}$  is defined as:

$$\mathcal{B}^\perp = \{\mathbf{x} : \mathbf{b} \cdot \mathbf{x}^T = 0, \forall \mathbf{b} \in \mathcal{B}\} = \{\mathbf{x} : \mathbf{G} \cdot \mathbf{x}^T = 0\} \quad (1.1)$$

The *parity check matrix*  $\mathbf{H}$  of a linear block code  $\mathcal{B}$  is the generator matrix of its dual code  $\mathcal{B}^\perp$ . As such, a codeword  $\mathbf{b}$  is in  $\mathcal{B}$  if and only if the matrix-vector product  $\mathbf{H}^T \mathbf{b} = 0$ . The rows of matrix  $\mathbf{H}$  consists of  $(n - k)$  linearly independent codewords from  $\mathcal{B}^\perp$  and each linearly independent codeword corresponds to a *parity-check equation*. A valid codeword should satisfy all parity-check equations. Since  $\mathbf{G}$  has the dimension  $k$  by  $n$ , the dimension of  $\mathbf{H}$  is  $(n - k) \times n$  when it is full rank. More coverage of linear block codes could be found in [7, 11].

### 1.3.4 Low-Density Parity-Check (LDPC) Codes

Low-density parity-check (LDPC) codes are a class of linear block codes with implementable decoders, which provide near-capacity performance on a large set of transmission channels [12, 4]. As its name indicates, the parity-check matrix of LDPC code is “sparse”; i.e., it contains only a small number of nonzero entries. The sparseness of  $\mathbf{H}$  limits the decoding complexity when code length is increasing.

Except for the sparseness of the parity check matrix  $\mathbf{H}$ , an LDPC code itself is no different from other block codes. It can be represented as  $(n, k)$ , where  $n, k$  stand for the length of the codeword  $\mathbf{u}$  and information bits  $\mathbf{b}$ , respectively. The corresponding full-rank  $\mathbf{H}$  has the

same dimension of  $(n - k) \times n$  as that mentioned above. LDPC codes are decoded iteratively based on the structure of the parity-check matrix, so the construction of  $\mathbf{H}$  is the major concern when designing LDPC codes with near-capacity performance.

The number of 1's in a column of parity-check matrix  $\mathbf{H}$  is called its *column weight*, which is denoted by  $d_{v,i}$  as the column weight of the  $i$ th-column. The *row weight* is the number of 1's in a row and  $d_{c,i}$  is defined as the row weight of the  $i$ th-row. When each column and each row of matrix  $\mathbf{H}$  has the same weight, i.e.,  $d_{v,i}$  and  $d_{c,i}$  have unique value for all columns and rows, the corresponding LDPC codes can be referred as  $(d_v, d_c)$  *regular* codes. On the opposite side, *irregular* LDPC codes have different values of column weight or row weight, which are specified by the *degree distributions* described below.

The construction method for LDPC codes proposed by Mackay and Neal is to add ones to satisfy the column weight constraint  $d_v$ . The location of the 1 entries are chosen randomly from the rows which have less than the  $d_c$  ones [4]. If there are rows with more positions unfilled and there are columns remaining to be added, choose the entry position randomly so that the row and column weight are equal or less than designated. The process can be started again or backtracked by a few columns until the correct row weight distribution is satisfied. An example of a length-12 regular MacKay Neal parity-check matrix with weight distribution  $(d_v = 3, d_c = 4)$  is:

$$\mathbf{H}_1 = \left[ \begin{array}{ccc|ccc|ccc} 1 & & & & 1 & 1 & & 1 & & \\ & 1 & & 1 & & & & & 1 & \\ & & 1 & & 1 & 1 & & 1 & & \\ & & & 1 & & & & & 1 & 1 \\ & & & & 1 & 1 & & & & 1 \\ & 1 & & & 1 & & & 1 & 1 & \\ & & 1 & & & 1 & & 1 & & \\ & & & 1 & 1 & & & & 1 & \end{array} \right] \quad (1.2)$$

## Graphical representation of LDPC codes

After several decades of ignorance since LDPC codes had been invented by Gallager, an important breakthrough in the research of LDPC codes is that Tanner [13] used a bipartite graph to provide a graphical representation of the parity-check matrix. A bipartite graph is a graph in which nodes can be partitioned into two subsets such that there are no edges connecting nodes within the same subset. For the parity-check matrix of an LDPC code, the two subsets are referred to as *variable nodes* and *check nodes*:  $n$  vertices for the variable nodes (or bit nodes),  $m$  vertices ( $m = n - k$ ) for the check nodes. Each row of the parity-check matrix corresponds to one check node and each column corresponds to one variable node. It follows that check-node  $i$  is connected to variable-node  $j$  if the element  $(i, j)$  of the  $\mathbf{H}$  matrix is equal to 1, i.e.,  $h_{ij} = 1$ . The Tanner graph corresponding to the parity-check matrix  $\mathbf{H}$  denoted by (1.2) is shown in Figure 1.6. In the graph, the number of edges incident upon a node is called the *degree* of the node, so a bipartite graph of a  $(d_v, d_c)$  LDPC code contains  $n$  variable nodes of degree  $d_v$  and  $m$  check nodes of degree  $d_c$ .

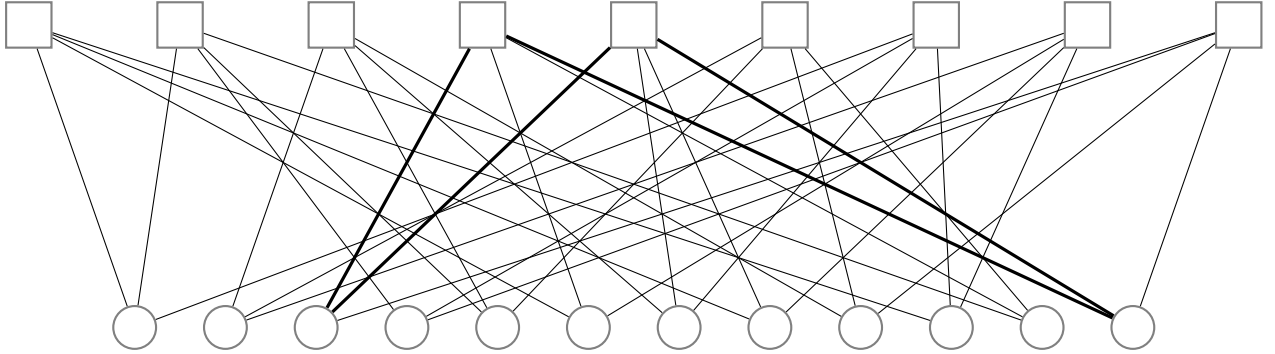


Figure 1.6: Tanner Graph corresponding to  $\mathbf{H}_1$ , the squares represent check nodes and the circles represent check nodes.

A *cycle* in a Tanner graph is a sequence of connected vertices which start and end at the same vertex in the graph and contain other vertices no more than once. The length of a cycle is the number of edges it contains. A bipartite graph only contains even-length cycles and the minimum length of a cycle contained in the graph is called the *girth*, which determines the performance of LDPC codes to some extent.

Since the parity-check matrix can be deduced from the Tanner graph, we can analyze the

properties of Tanner graph to generate some useful conclusions. The LDPC decoding can also be performed by using Tanner graph. After the major parameters of the Tanner graph are decided, which include the degrees of the variable and check nodes and the girth of the graph, we can still choose how the vertices of the graph are connected with each other.

Irregular LDPC code have different weight distribution in the columns or in the rows, thus they cannot be defined solely in terms of the weight (degree) parameters  $d_v$  and  $d_c$  [14]. We can instead use *degree distributions* or *node distributions* to describe the variety of node degrees in the Tanner graph. A degree distribution can be described by a polynomial with the form:

$$\gamma(x) = \sum_{i \geq 2} \gamma_i x^{i-1}, \quad (1.3)$$

where  $\gamma(x)$  has nonnegative coefficients and  $\gamma(1) = 1$ . The use of polynomials leads to elegant and compact descriptions of the future results.  $\gamma_i$  denotes the fraction of edges in the graph which are connected to a node of degree  $i$ . Thus an irregular code ensemble with length  $n$  can be fully described by degree distribution  $\lambda$  and  $\rho$ :

$$\lambda(x) = \sum_{i \geq 2}^{d_v} \lambda_i x^{i-1}$$

where  $\lambda(x)$  specifies the variable-node degree distribution, which is the fraction of edges connected to variable nodes of degree  $i$ . Similarly, the check node degree distribution could be written as,

$$\rho(x) = \sum_{i \geq 2}^{d_c} \rho_i x^{i-1}$$

where  $\rho(x)$  is the check-node degree distribution, which is the fraction of edges connected to check nodes of degree  $i$ . Regular  $(d_v, d_c)$  codes can also be represented by polynomials; e.g, for the  $(3, 6)$  regular code, we have  $\lambda(x) = x^2$  and  $\rho(x) = x^5$ .

Let  $n$  be the number of variable nodes and  $m$  be the number of check nodes. The check-node and variable-node degree distributions are related by the following set of equations:

$$m \cdot \int_0^1 \lambda(x) dx = n \cdot \int_0^1 \rho(x) dx \quad (1.4)$$



$$r(\lambda, \rho) = \frac{n - m}{n} = 1 - \frac{\int_0^1 \rho(x) dx}{\int_0^1 \lambda(x) dx}, \quad (1.5)$$

where  $r$  is the LDPC code rate depending on the degree distribution. Equation (1.5) is valid when all the check equations are linearly independent, in other words, the corresponding parity-check matrix is full rank. We can also define node-perspective degree distribution as:

$$\begin{aligned} \Lambda(x) &= \sum_{i \geq 1}^{d_v} \Lambda_i x^i \\ P(x) &= \sum_{i \geq 1}^{d_c} P_i x^i \end{aligned}$$

where  $\Lambda_i$  is the fraction of variable nodes of degree  $i$  and  $P_i$  is the fraction of check nodes of degree  $i$ . We can easily deduce the relationship between the edge degree distribution and node degree distribution as:

$$\begin{aligned} \Lambda_i &= \frac{\lambda_i/i}{\int_0^1 \lambda(x) dx} = \frac{\lambda_i/i}{\sum_{j \geq 2} \lambda_j/j} \\ P_i &= \frac{\rho_i/i}{\int_0^1 \rho(x) dx} = \frac{\rho_i/i}{\sum_{j \geq 2} \rho_j/j} \end{aligned}$$

## Decoding Algorithm

LDPC codes can be decoded using an iterative decoding algorithm whose complexity grows linearly with the length of the code. Codeword  $\mathbf{u}$  is transmitted over the channel with the disturbing noise, and then the decoder will recover the information bits according to the received codeword. The sum-product decoding algorithm is a soft decision message-passing algorithm, which accepts the probability of each received bit as input [15]. The Tanner graph could be used to help understand the iterative decoding process, which could be treated as an exchange of messages between variable nodes and check nodes through the connected edges.

The input probabilities are called *a priori* probabilities for the received bits; the output probabilities of the decoder are called *a posteriori* probabilities because they are the conditional probability of the transmitted bits given the received bit probability or related evidence. In the sum-product message-passing decoding algorithm, all messages are assumed to

be loglikelihood ratios (LLR), i.e., of the form  $L(u_i) = \log(p(0)/p(1))$ . If  $p(u_i = 0) > p(u_i = 1)$  then  $L(u_i)$  is positive and the greater the magnitude the more possible that  $u_i = 0$ . The benefit of using logarithmic ratios is that they can transform the multiplication of probabilities into additions, which can clearly reduce implementation complexity.

The decoding process is to compute the maximum a posteriori probability for each codeword bit,  $P_i = P(u_i = 1|N)$ , which is the probability that the  $i$ -th codeword bit is a 1 conditioned on the event  $N$  that all parity-check constraints are satisfied. Belief propagation on a graph without cycles can be closely approximated by the MAP algorithm.

Extrinsic information is the message sent from a node  $n$  along an edge  $e$  and cannot depend on any message previously received on edge  $e$ . The notation  $\mathcal{M}(i)$  is the set of check nodes connected to variable node  $i$ , i.e., the positions of 1s in the  $n$ -th column of  $\mathbf{H}$ . The set  $\mathcal{N}(j)$  contains variable nodes that participate in the  $m$ -th parity-check equation, i.e., the positions of 1s in the  $m$ -th row of  $\mathbf{H}$ . Let  $\mathcal{N}(j) \setminus i$  represents exclusion of  $i$  from the set  $\mathcal{N}(j)$ . The message from variable node  $i$  to check node  $j$  is denoted as  $Z_{i \rightarrow j}$ , which is based on the messages from all the check nodes connecting to variable node  $i$  except the check node  $j$  itself. Another notation  $Z_i$  is the a posteriori message of variable node  $i$ , which can be used for the hard decision. Similarly,  $L_{j \rightarrow i}$  denotes the message from the  $j$ -th check node to the  $i$ -th symbol node based on all the variables checked by  $j$  except  $i$ .  $\mathbf{y} = [y_1, y_2, \dots, y_n]$  denote the received codeword. Omitting the proof, the iterative sum-product message-passing decoding algorithm [16] is listed as follows:

Step 1: Initialize  $Z_i^{(0)}$  of each variable node  $i$  based on a posteriori LLR received from the channel. In the case of additive white Gaussian noise (AWGN) channel,  $Z_i^{(0)} = \frac{2}{\sigma^2} y_i$  for each variable node, where the  $\sigma^2$  is the variance for the additive noise.

Step 2: Variable nodes send  $Z_{i \rightarrow j}$  to each check node  $j \in \mathcal{M}(i)$ .

Step 3: Check nodes connected to variable node  $i$  send

$$L_{j \rightarrow i}^{(l)} = 2 \tanh^{-1} \left( \prod_{i' \in \mathcal{N}(j)/i} \tanh \left( \frac{Z_{ji'}^{(l-1)}}{2} \right) \right) \quad (1.6)$$

Step 4: For each variable node  $i$ , compute the messages sent to check nodes  $j$  and its

own decision message  $Z_i^{(l)}$  at the  $l$ -th iteration:

$$Z_{i \rightarrow j}^{(l)} = \sum_{j' \in \mathcal{M}(i)/j} L_{j'i}^{(l)} + L_n^{(0)} \quad (1.7)$$

$$Z_i^{(l)} = \sum_{j \in \mathcal{M}(i)} L_{ji}^{(l)} + L_n^{(0)} \quad (1.8)$$

Step (2) and step (4) are repeated until the end of the decoding. The decoding process could be stopped early if the correct codeword is found or a preset maximum number of iterations is achieved.

## 1.4 Satellite Communication

Satellite communication plays a vital role in the global telecommunications by using artificial satellites to provide communication links between various points on Earth. There are approximately 2000 artificial satellites orbiting Earth and they are responsible for relaying signals carrying voice, video and data to and from one or multiple locations worldwide. Satellite communications can complement existing terrestrial infrastructure, providing competitive advantages such as greater coverage area, terrestrial-free network and the higher bandwidth. The limited access points also make satellite communications secure and private. But the high cost of launching satellites and the propagation delay are also the disadvantages for the satellites communication.

The two main components of satellite communication are the satellite itself, known as the *space segment* and the *ground segment*, which consists of fixed or mobile transmission, reception and ancillary equipments. The primary task of a satellite is to receive signals from a ground station and retransmit them down back to earth at a different frequency. Some satellites also have the capability to process the received data before the retransmission. The data could be recovered first and then modulated using a different frequency or at a different data rate. The *uplink* indicates the process of an earth station transmits the signal up to the satellite. The terrestrial data may need to pass a baseband processor, a up converter, a high powered amplifier and through a parabolic dish antenna up to the

designated satellite. The retransmission from the satellite to the ground station is termed the *downlink* which works in the reverse fashion as the uplink.

The satellites could be categorized according to their service types; the fixed service satellites are responsible for the point to point communication, the broadcast service satellites are used for the satellite television or radio, the mobile service satellites are mainly for the satellite phone service. Different satellites use different frequency bands. The  $K_u$  band, which ranges from 12-18 GHz, is used for fixed and broadcast services. The frequency range from 26.5 to 40 GHz  $K_a$  band is used for point to point communications.

### 1.4.1 DVB-S2 Standard

Digital Video Broadcasting (DVB) is a suite of internationally accepted open standards for digital television and radio broadcasting. The family of DVB standards are maintained by a Joint Technical Committee (JTC) of European Telecommunications Standards Institute (ETSI), European Committee for Electrotechnical Standardization (CENELEC) and European Broadcasting Union (EBU). DVB systems distribute data using a variety of approaches, including:

- Satellite: DVB-S, DVB-S2 and DVB-SH
- Cable: DVB-C, DVB-C2
- Terrestrial television: DVB-T, DVB-T2
- Microwave: DVB-MT, DVB-MC and DVB-MS

These standards define the physical layer and data link layer and the main difference among them are the modulation schemes and error correcting codes. For the standards used for the same purpose, the second generation standard normally outperforms the previous generation one by using adaptive modulation, more powerful channel coding and other advanced strategies. For instance, DVB-T is the first generation DVB standard for the transmission of digital terrestrial television. The second generation DVB-T2 adds the 256-QAM modulation and upgrades the error correct code from the combination of convolutional coding and Reed

Solomon code to the combination of BCH and LDPC code. The same evolving trend can also be found from the DVB-S2 and DVB-C2.

Digital Video Broadcasting - Satellite (DVB-S) is the original DVB standard for the transmission of satellite television and data. Its applications include the services in direct broadcast satellite like the Dish Network in the U.S. and Bell TV in Canada. Digital Video Broadcasting - Satellite - Second Generation (DVB-S2) is a digital television broadcast standard that has been designed as a successor for the previous popular DVB-S system. It was developed in 2003 by the DVB Project. The standard is based on, and improves upon DVB-S by incorporating new coding and modulation schemes. The adaptive coding and modulation mode also make better use of the bandwidth under different transmission conditions. DVB-S2 is designed for broadcast services including standard and HDTV (High-definition television) signals, interactive services, and professional data content distribution. A better performance of about 30% gain is achieved by DVB-S2 than the previous DVB-S standard at the same transmission bandwidth and signal power.

### 1.4.2 APSK

Amplitude Phase Shift Keying (APSK) modulation transmits a data sequence by changing both the amplitude and the phase of the carrier signal. It can be considered as a combination of phase-shift keying and amplitude-shift keying (ASK). It normally arranges the symbols into two or more concentric rings with a constant phase offset  $\theta$ . For example, 16-APSK uses a double ring PSK format, this is called 4 – 12 16-APSK with four symbols in the center ring and 12 in the outer ring, which is shown in Fig. 1.7.

APSK is an attractive modulation scheme for nonlinear satellite channels due to its power and spectral efficiency together with the robustness against nonlinear distortion. The concept of APSK modulation and its suitability for nonlinear channels was proposed in [17]. At that time, it was concluded that PSK performs better than APSK modulation over nonlinear channel with single carrier operation. However, based on the results from [18], this conclusion was reverted and APSK became a popular modulation scheme for advanced satellite communications. The commercial applications of APSK include DVB-S2, DVB-SH

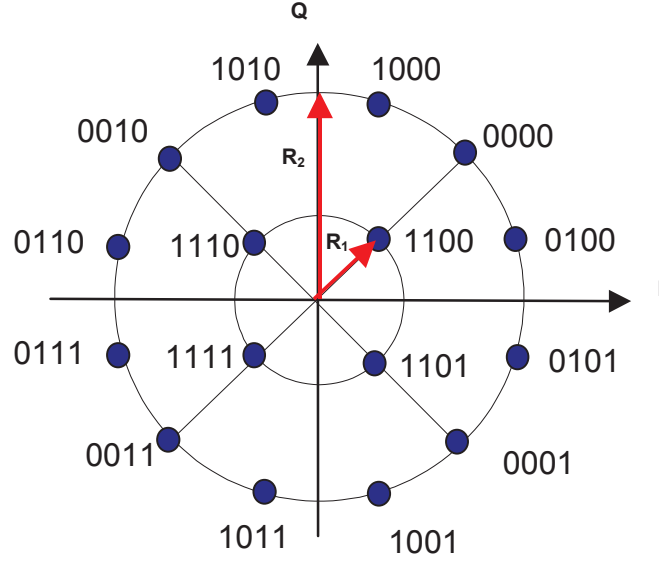


Figure 1.7: 16-APSK

(Digital Video Broadcasting via Satellite to Handheld devices) and IPoS (Internet Protocol over Satellite), etc.

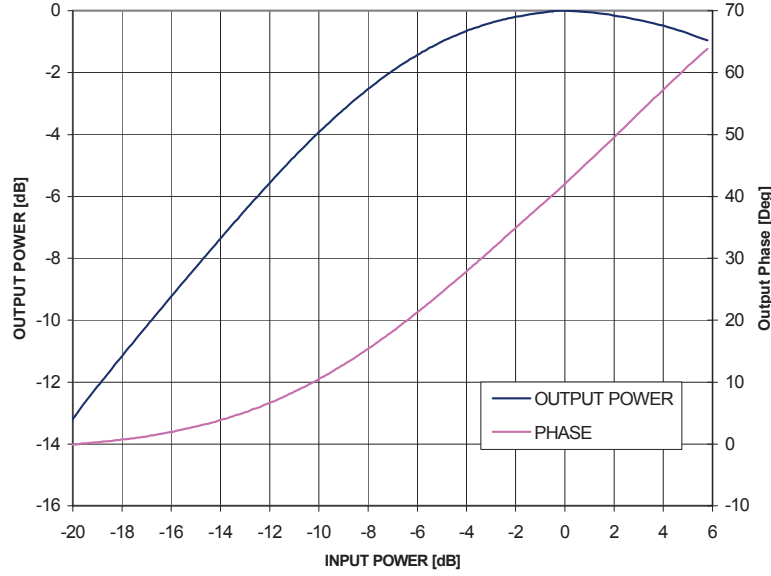
### 1.4.3 TWTA

A traveling-wave tube (TWT) is a specialized vacuum tube that is used to provide power amplification for signals with high frequency. It is usually integrated as part of an assembly known as a traveling-wave tube amplifier (TWTA), which is normally used as the amplifier in satellite transmitter. TWTAs provide high power amplification for the input weak signals, while it also cause side effects along with their efficient amplification performance.

There are several aspects to specify the characteristics of a TWTA:

- Peak Power
- AM/AM Transfer and AM/PM Conversion curves
- Input Back-off (IBO) vs. Output back-off (OBO)
- Gain

The curves of output power as the function of the input power for TWTAs are also called AM/AM transfer curves. The blue curve in Fig. 1.8 shows the AM/AM transfer

Figure 1.8: Non-Linearized TWTA characteristic<sup>1</sup>

characteristic of a  $K_a$  band (frequency range from 26.5-40 GHz) TWTA defined in DVB-S2 standard [19]. The output increase as the increasing of input power until the input power reaches certain region. At a point called *saturation*, any further increase in input power produces no further increase in output power. The saturate point or region is designed at input power level of 0 dBm for most TWTA. Channel amplifiers are often used before the TWTA to assure the input power level is either near the saturation or at some other pre-determined input power values.

The power level that the TWTA is actually being operated is called the operating point or *back-off*. The operating point of the input power is called *Input Back-off* (IBO), and *Output Back-off* (OBO) means the corresponding output power value. Generally the OBO attracts more attention in the satellite operation terms since the operator is more concerned about the total power generated out of the transponder. Theoretically the TWTA should be operated at the saturation point to achieve the maximum output power, but the non-linear behavior when operating around the saturation point or the near region causes significant signal distortions. So the real operation point is determined individually under different circumstances.

<sup>1</sup>This figure comes from DVB-S2 standard appendix H.7

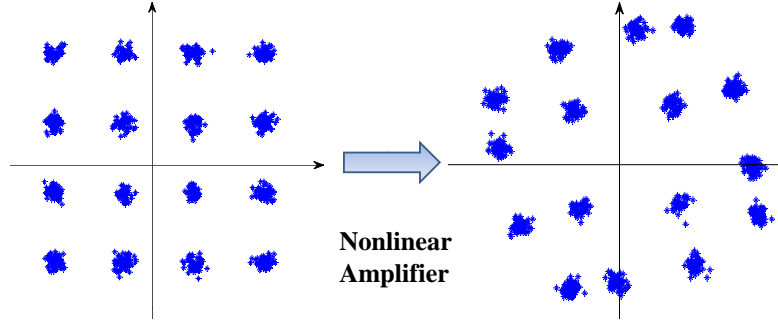


Figure 1.9: Distorted 16-QAM on the outputs of non-linear TWTA

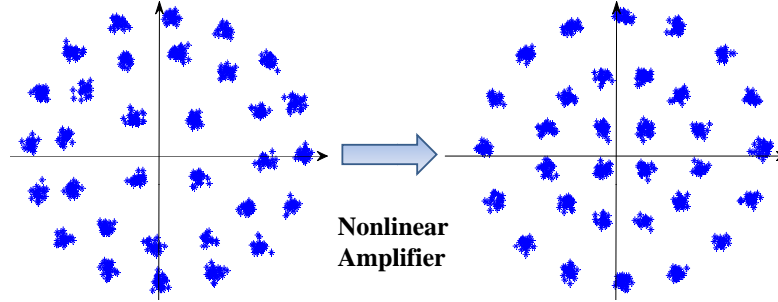


Figure 1.10: Distorted 32-APSK on the outputs of non-linear TWTA

Aside from the amplitude transfer curves, phase shift properties are also a key characteristic provided by the TWTA manufacturers. The phase shift is another cause of degradation because of the non-linearity of the amplitude. The phase distortion depends on the drive level, i.e., the input power level. The curve of the phase shift versus the input back off is shown as a pink curve in Fig. 1.8. The two characteristic curves in that figure are all for  $K - a$  band TWTA for the DVB-S2 standard. The original figure could be found in the appendix of [19]. The phase shift curve can be called the AM/PM conversion curve. The degradation of the phase shift is about 0.6 dB and the non-linear amplitude could effect up to 2.5 dB of the signal to noise ratio [20].

To better understand the impact of nonlinear characteristic of TWTA, two examples corresponding to 16-QAM and 32-APSK are shown in Fig. 1.9 and 1.10. The sub-figures in the left show the constellation symbols with additive noise. The noise variance is set to be low to keep the constellation recognizable. The sub-figures in the right are the distorted symbols through non-linear TWTA, which shows both the amplitude and phase impact generated by the TWTA. From these two figures, APSK symbols preserve a good shape while the positions



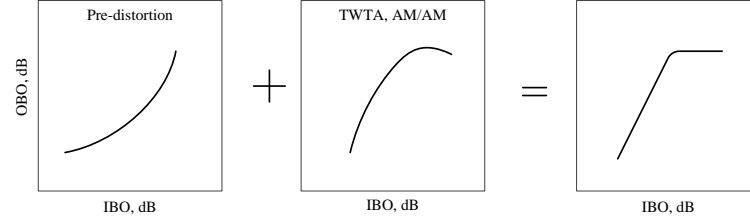


Figure 1.11: Magnitude Predistortion linearizers generate a response opposite to a TWTA's characteristics

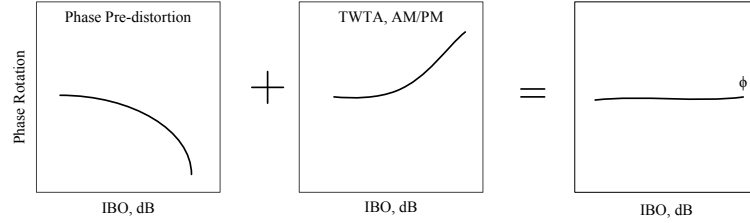


Figure 1.12: Phase Predistortion linearizers generate a response opposite to a TWTA's characteristics

of QAM symbols are significantly changed under the non-linear distortion. Except adopting modulations schemes like APSK, the amplifier non-linearity impairment can be reduced by using a technique called *pre-distortion*.

Pre-distortion is one of the strategies to compensate the nonlinearities of the amplifier. The principle of pre-distortion is to distort the signals before transmitting to the amplifier, the distortion operation is achieved by an additional device whose characteristics are close to the inverse of those of the amplifier. The detail about the linearizing of high power amplifier could be found in [21, 22, 23, 24, 25].

A typical amplitude transfer characteristic of the TWTA is shown in the middle part of Fig. 1.11, the sum of the characteristic from the predistorter and the TWTA would cancel out the non-linearity of the TWTA and the resulted linear AM/AM transfer curve is shown in the rightmost figure. The same operation to mitigate the phase distortion caused by the TWTA can also be achieved by the pre-distortion 1.12. The predistorter, or the linearizer, has to match the characteristics of the TWTA in order to produce the maximum beneficial effects. The algorithms to implement the pre-distortion strategy can be found in [26, 27]. Fig. 1.13 shows the linearized constellation symbols of 16-QAM and 32-APSK after passing through the predistortion followed by the TWTA.

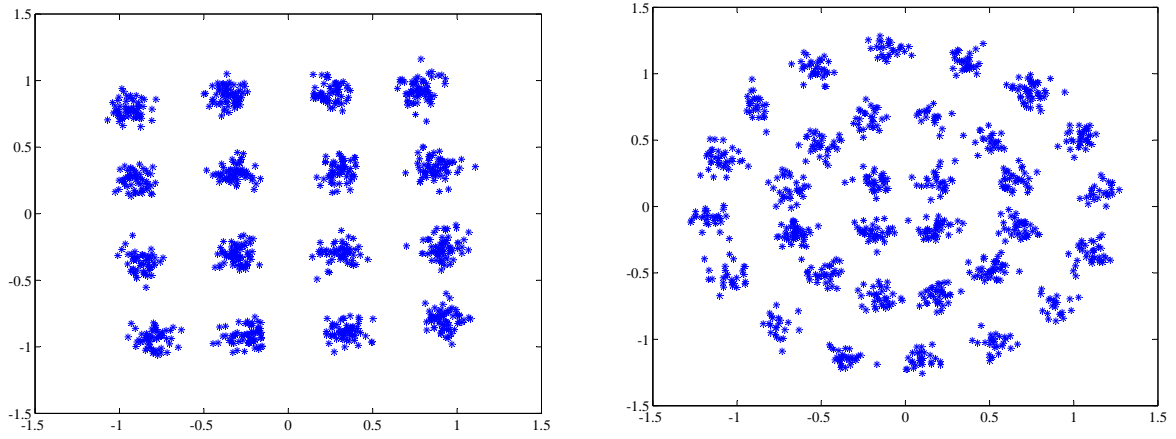


Figure 1.13: Predistorted QAM and 32-APSK on the outputs of non-linear TWTA

The benefits of a linearized TWTA are obvious: the linear region can be expanded and cause smaller non-linear effects. The disadvantages include the requirement of additional power and the additional weight of the extra equipment, the difficulty of tuning and not suitable for all traffic types.

#### 1.4.4 DVB-S2 Performance

The original standard DVB-S uses QPSK modulation with concatenated convolutional and Reed-Solomon codes. The evolved DVB-S2 chose the enhanced modulation schemes and low density parity check (LDPC) codes which actually provided more than 35% throughput improvement with respect to the previous DVB-S standard. For the LDPC codes defined in DVB-S2 standard, certain structure is imposed on parity check matrices to facilitate the codes description and easy encoding. More specifically, the parity check matrix  $\mathbf{H}$  is restricted to be the following form:

$$\mathbf{H}_{(N-K) \times N} = [\mathbf{A}_{(N-K) \times K} \mathbf{B}_{(N-K) \times (N-K)}] \quad (1.9)$$

Where  $\mathbf{B}$  is dual diagonal sub-matrix:

$$\mathbf{B}_{n-k,n-k} = \begin{bmatrix} 1 & & & & & \\ & 1 & 1 & & & \\ & & 1 & 1 & & 0 \\ & & & 1 & & \\ & 0 & & & \ddots & \ddots \\ & & & & & 1 \\ & & & & & 1 & 1 \end{bmatrix} \quad (1.10)$$

Then the parity bits can be solved recursively based on the speciality structure of the dual diagonal matrix. The code with such structure parity-check matrix is also known as an *extended irregular repeat accumulate* (eIRA) code [28].

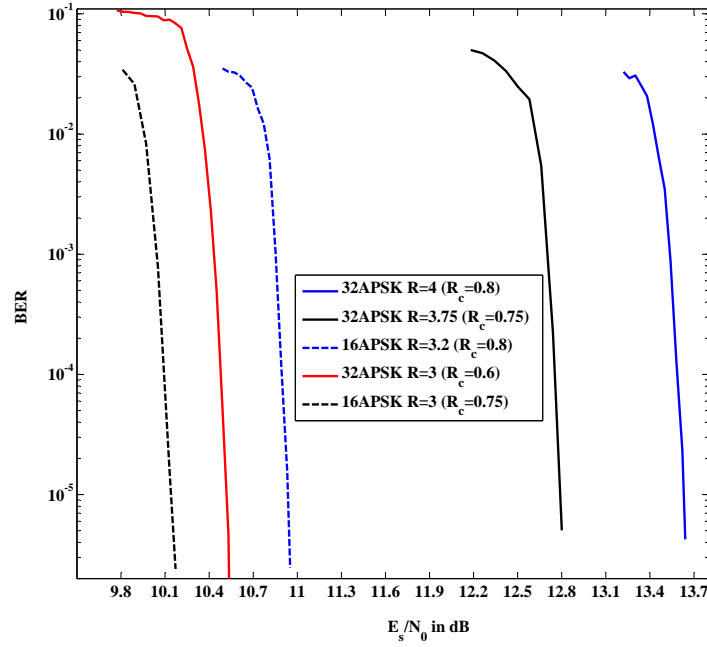


Figure 1.14: Performance of DVB-S2 standardized LDPC coded over AWGN channel

In DVB-S2, a wide range of bandwidth efficiencies from 0.5 bits per symbol up to 4.5 bits per symbol is covered by defining ten different code rates  $1/4, 1/3, 1/2, 3/5, 2/3, 3/4, 4/5, 5/6, 8/9$  and  $9/10$  with four different modulation schemes QPSK, 8-PSK, 16-APSK, 32-APSK. The error performances of the uncoded modulation schemes are shown in Fig. 1.2. 32-APSK has the worst error performance while it could transmit the most number of bits per symbol. The LDPC coded block length can be either  $N = 16\,200$  (short) or  $N = 64\,800$  (normal) bits

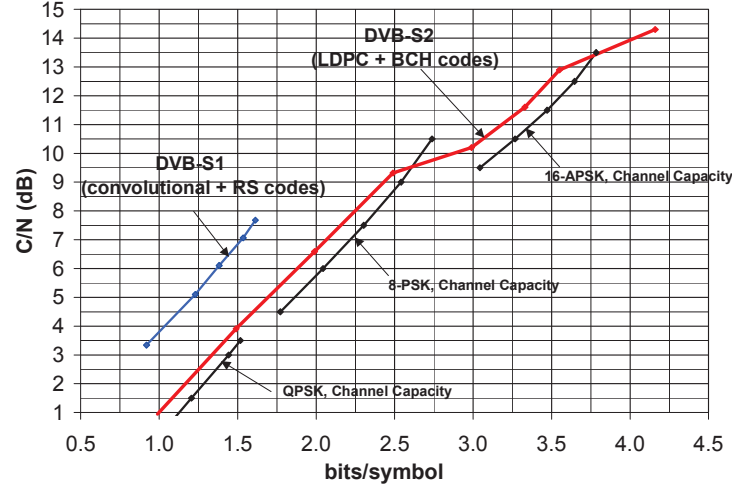


Figure 1.15: Comparison of DVB-S2 Codes to DVB-S and Channel Capacity<sup>2</sup>

for all rates. To improve the performance, irregular LDPC codes are used where degrees of variable nodes are different.

Performance of various code rates with 16-APSK and 32-APSK constellations on AWGN channel is depicted in Figure 1.14. Maximum number of decoder iterations is set to be 100. If no valid codeword is found then the current decoder outputs are estimated at the end of 100-th iterations.

For comparison, the performance of DVB-S code and Shannon limits of constellations are provided in [19] and cited as Fig. 1.15. It is important to note that the system operation parameters of DVB-S2 are chosen by following the theoretical limit. Compared to the concatenated convolutional and Reed-Solomon codes in DVB-S, the improvement of more than 35% is achieved by using the modern channel coding strategy [29].

## 1.5 Thesis Outline

The objective of the dissertation is to improve the performance of systems like DVB-S2 that use APSK modulation and LDPC coding. Several strategies including constellation shaping, LDPC coding optimization, modulation optimization are used to push the system to approach the theoretical limit.

---

<sup>2</sup>the original figure come from [19]

The dissertation is divided into 5 chapters. Chapter 2 covers the channel capacity concepts and the idea of constellation shaping. Chapter 3 introduces the shaping system with 16-PAM and 32-APSK modulation. Both turbo codes and LDPC codes are used as the channel coding scheme to test the constellation strategy. A useful tool named EXIT charts is reviewed in Chapter 4 and used to optimize the degree distribution of LDPC codes. Constellation symbol mapping is investigated to find better rules than the DVB-S2 standard. Bit error rate results indicate the performance improvement by combining better mapping rules with optimized LDPC codes.

## Chapter 2

# Channel Capacity and Constellation Shaping

In a digital communication system, reliable transmission requires that the useful source signal be reconstructed at the sink with an arbitrarily low probability of error. The maximum transmission rate at which reliable transmission can be satisfied is called the *channel capacity* [30]. This chapter begins with a discussion of the concept of capacity. A numerical method for calculating capacity value is presented and compared with results generated by *Monte-Carlo simulation*. In addition, a simple method to change the input symbol distribution called *constellation shaping* is introduced as a strategy to better approach the capacity limit under modulation constraint. Next, the system model including constellation shaping, i.e., *shaped system* is described. Several examples are given to help understand the shaping strategies. The optimized parameters for shaped systems with PAM and APSK modulation are found based on mutual information analysis. The *peak to average power ratio* for shaped system is also considered and the summary is given at the end of this chapter.

### 2.1 Channel Capacity

As the upper bound on the amount of reliably transmitted information, channel capacity can also be interpreted as a theoretical criteria that should be maximized given the system constraints such as modulation types, channel conditions etc. Channels with additive white

gaussian noise (AWGN) are the main focus of this chapter. The derivation of channel capacity is reviewed in the beginning of this section, then Monte-Carlo simulation and *Gaussian-Hermite quadrature* are introduced as methods for calculating the capacity. Several capacity examples are given after introducing two coded modulation system models, which are called *bit-interleaved coded modulation* (BICM) and *bit-interleaved coded modulation with iterative decoding* (BICM-ID) system.

### 2.1.1 The Unconstrained Capacity

The *discrete channel* is a system consisting of an input alphabet  $X$ , an output alphabet  $Y$  and a probability transition probability for each pair of input symbol  $x_i$  and output symbol  $y_i$ . The transition probability  $p(y_i|x_i)$  is the conditional probability of receiving output symbol  $y_i$  given that  $x_i$  is transmitted. Only *memoryless* channels are considered in this dissertation, in which the output probability distribution only depends on the channel input at the same time. The model for a discrete-time AWGN channel is,

$$y_i = x_i + z_i, \quad (2.1)$$

where  $y_i$  is the channel output at time  $i$ . The output of the channel is the summation of the channel input and additive Gaussian noise. The input signal  $x_i$  is assumed to be independent of noise  $z_i$ , which is Gaussian distributed with zero mean and variance  $N$ . The mutual information  $I(X;Y)$  can be expanded as:

$$\begin{aligned} I(X;Y) &= H(Y) - H(Y|X) \\ &= H(Y) - H((X + Z)|X) \\ &= H(Y) - H(Z|X) \\ &= H(Y) - H(Z). \end{aligned} \quad (2.2)$$

The channel capacity is defined as the maximum value of mutual information between the channel input and output over all possible input distribution  $p(X)$ . The input of the channel is normally power limited, otherwise it may lead to infinite channel capacity. The capacity

of Gaussian channel with power constraint  $P$  is:

$$C = \max_{p(x), E(X^2) \leq P} I(X; Y), \quad (2.3)$$

Since  $Z$  is Gaussian distributed noise and independent of  $X$ , the entropy of which is  $H(Z) = \frac{1}{2} \log 2\pi e N$  according to the entropy definition [1]. The entropy of the output  $Y$  is bounded by  $\frac{1}{2} \log 2\pi e(P + N)$  based on the theorem that the normal distribution with given variance maximized the entropy. The notation  $P$  is the power constraint of the channel input  $X$  and  $N$  is the noise variance [1]. Equation (2.2) can be written as:

$$I(X; Y) \leq \frac{1}{2} \log 2\pi e(P + N) - \frac{1}{2} \log 2\pi e N = \frac{1}{2} \log \left( 1 + \frac{P}{N} \right) \quad (2.4)$$

The maximum mutual information  $I(X; Y)$  is achieved when  $Y$  is Gaussian distributed, the input  $X$  should be also Gaussian since the noise follows Gaussian distribution. The units of capacity in the form of equation (2.4) is bits per transmission or bits per channel use. Channel capacity is the upper bound on the transmission rate for reliable communication over the noisy channel. If the bit transmission rate is greater than capacity  $C$ , it is not possible to recover the information bits with arbitrary small error probability.

The channel with waveform inputs and outputs is also called *continuous time channel* or *waveform channel* with model of  $Y(t) = X(t) + Z(t)$ . Assume that the channel has limited bandwidth  $W$  and the channel noise is additive white Gaussian noise. This continuous-time AWGN channel can be reduced to an equivalent discrete-time AWGN channel according to the sampling theorem and the theorem of irrelevance [1]. The continuous input and output signal could be represented by discrete samples spaced by Nyquist's sampling rate. The continuous-time signal with bandwidth  $W$  during time interval  $[0, T]$  can be completely determined by  $2WT$  samples. The signal power per sample is  $P/2W$  and the noise power per sample is  $N_0/2$ , where  $N_0$  is the noise power per unit of bandwidth. It follows that the channel capacity per sample can be written as:

$$C = \frac{1}{2} \log \left( 1 + \frac{P}{\frac{N_0}{2}} \right) = \frac{1}{2} \log \left( 1 + \frac{P}{N_0 W} \right). \quad \text{bits per sample} \quad (2.5)$$

Since there are  $2W$  samples per second, the capacity can also be expressed in units of bits



per second:

$$C = W \log \left( 1 + \frac{P}{N_0 W} \right) \quad \text{bits per second} \quad (2.6)$$

The capacity can be more informative when described as the function of bit energy  $\mathcal{E}_b$

$$C = W \log \left( 1 + \frac{P}{N_0 W} \right) \quad (2.7)$$

$$= W \log \left( 1 + \frac{\mathcal{E}_b r_b}{N_0 W} \right) \quad \text{bits/sec.} \quad (2.8)$$

where  $r_b$  is the data rate. There are several interesting results that can be obtained by replacing  $r_b$  with  $C$  and treating  $C/W$  as the normalized channel capacity with units of bits per second per Hz.

$$\frac{C}{W} = \log \left( 1 + \frac{C}{W} \frac{\mathcal{E}_b}{N_0} \right)$$

The signal to noise ratio  $\mathcal{E}_b/N_0$  should increase exponentially to make  $C/W$  approach infinity. The minimum  $\mathcal{E}_b/N_0$  is  $-1.6$  dB as  $C/W$  getting close to 0.

Fading channels are characterized as having a random fluctuation in the received amplitude. For such channels, the concept of capacity can be extended to the so-called *ergodic capacity*. Fading channels can be modeled as  $Y_i = h_i \cdot X_i + Z_i$ , where  $h_i$  is the fading coefficient. A simple fading model is *ergodic Rayleigh fading* channel, in which the fading coefficient  $h_i$  is independent and identically distributed (i.i.d.) Rayleigh distributed and the channel information is known at the receiver for every time instant. Let  $\hat{P}$  denote the average transmit signal power,  $N_0/2$  the noise power spectral density, and  $W$  the received signal bandwidth. The instantaneous received SNR is then  $\hat{P}h[i]/N_0B$  and its expected value is  $\hat{P}\hat{h}/N_0W$ . Since  $\hat{P}/N_0W$  is a constant, the distribution of fading coefficients  $h[i]$  determines the distribution of SNR. To find the capacity, we must average over the distribution of  $h[i]$ , i.e.,

$$\begin{aligned} C &= E_{|h[i]|^2} \left[ W \log_2 \left( 1 + \frac{\hat{P}|h[i]|^2}{N_0 W} \right) \right] \\ &= \int_{-\infty}^{\infty} W \log_2 \left( 1 + \frac{\hat{P}x}{N_0 W} \right) p(x) dx \\ &= \int_{-\infty}^{\infty} W \log_2 (1 + \beta) p(\beta) d\beta \end{aligned} \quad (2.9)$$

where  $\beta = \frac{\hat{P}_x}{N_0 W} = \frac{\hat{P}|h|^2}{N_0 W}$  is the instantaneous SNR and  $p(\beta)$  is the SNR distribution. The capacity expression of (2.9) is the ergodic capacity of the fading channel [31].

### 2.1.2 Capacity under Coded Modulation

Notice that the expressions in equations (2.4), channel capacity is attained when the input signal is Gaussian distribution, i.e.,  $\mathbf{X} \sim (\mathbf{0}, P)$ . However, the input symbols are normally selected from a finite alphabet  $\mathbf{X} = \{x_0, x_1, \dots, x_{M-1}\}$ . The output of the channel is assumed to be any value on the real line, i.e.,  $\mathbf{Y} \in (-\infty, \infty)$ . When the channel input is  $x_k$  and noise  $Z$  is zero-mean Gaussian random variable with variance  $\sigma^2$ , the channel output  $Y$  follows a Gaussian distribution with mean  $x_k$  and variance  $\sigma^2$ . The transition probability density function  $p(y|x)$  is:

$$p(y|X = x_k) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(y-x_k)^2/2\sigma^2}. \quad (2.10)$$

The mutual information between discrete  $X$  and continuous  $Y$  is also called *average mutual information*, which is defined as:

$$\begin{aligned} I(X; Y) &= E[i(X; Y)] \\ &= E \left[ \log \frac{p(x, y)}{p(x)p(y)} \right] \\ &= E \left[ \log \frac{p(y|x)}{p(y)} \right]. \end{aligned} \quad (2.11)$$

the expectation in equation (2.11) is averaged with respect to the *joint probability density function*  $p(x, y)$ . The capacity is the maximum average mutual information between the discrete input  $\mathbf{X} = \{x_0, x_1, \dots, x_{M-1}\}$  and the output  $\mathbf{Y} \in \mathbb{R}$ , which can be written like,

$$C = \max_{P(x_i)} \sum_{i=0}^{M-1} \int_{-\infty}^{\infty} p(y|x_i) p(x_i) \log_2 \frac{p(y|x_i)}{p(y)} dy, \quad (2.12)$$

where

$$p(y) = \sum_{k=0}^{M-1} p(y|x_k) p(x_k). \quad (2.13)$$

Capacity in equation (2.12) has units of bits per symbol when a base-2 logarithm is used, which is also used in the following. The input modulation symbol  $x_k$  is normally selected

with equal probability, i.e.,  $p(x_k) = 1/M$ , in which case the transmitter and receiver are easy to implement. The mutual information with uniform input distribution is called *symmetric information rate* (SIR) and then (2.11) could be rewritten as:

$$\begin{aligned}
 I(X; Y) &= E_{p(x,y)} \left[ \log p(y|x) - \log \left( \sum_{x_k \in \mathbf{X}} p(y|x_k) p(x_k) \right) \right] \\
 &= E_{p(x,y)} \left[ \log p(y|x) - \log(1/M) - \log \left( \sum_{x_k \in \mathbf{X}} p(y|x_k) \right) \right] \\
 &= \log(M) - E_{p(x,y)} \left[ \log \frac{\sum_{x_k \in \mathbf{X}} p(y|x_k)}{p(y|x)} \right] \tag{2.14}
 \end{aligned}$$

Unfortunately, there is no simplified expression for the SIR expression in (2.14). Even after substituting the Gaussian transition probability  $p(y|X = x_k)$  shown in (2.10) into equation (2.14), numerical integration techniques must be applied. In the actual calculation, the expected value in (2.14) can be approximated by the sample mean of a sufficiently large sampling of the distribution for  $Y$ . The capacity under modulation constraint is limited by an upper bound  $(\log_2 M)$ , where  $M$  is the cardinality of modulation constellation. Fig. 2.1 shows the unconstrained Shannon capacity bound with unit bits per sec per Hertz. The SIR, or the maximum code rate under uniform input distribution, as a function of  $\mathcal{E}_b/N_0$  is also shown in Figure 2.1 for several modulation schemes. Since the SIR is achieved when the channel input follows uniform distribution rather than the maximized mutual information over all possible input distributions, the asymptotic loss between SIR and the Shannon capacity is up to 1.53 dB [32].

The capacity defined in equation (2.12) is called the *coded modulation* (CM) capacity, which is also the maximum transmission rate achieved with arbitrarily low error probability with the input signals constrained in finite set  $\mathcal{X}$ . The CM capacity corresponds to the capacity of the memoryless ‘CM channel’ in Fig. 2.2 for a given constellation  $\mathcal{X}$ .

BICM is introduced in [33] and implemented by concatenating a binary encoder with a modulator defined over a cardinality  $M$  signal set through a bit-level random interleaver. The output of the encoder is bit-interleaved to produce the sequence  $\mathbf{z}$ , which will be divided into multiple sub-sequences with length  $m = \log_2 M$  and each one of them is mapped to a modulated signal  $x_k, k \in 0, 1, \dots, M-1$ . The bit-interleaver is assumed to be ideal, i.e., the

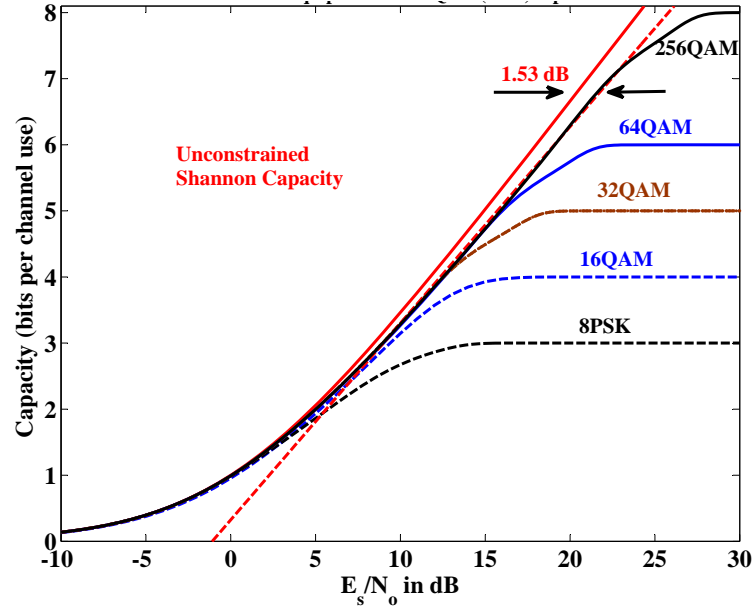


Figure 2.1: Unconstrained AWGN channel capacity and SIR with equiprobable M-QAM(PSK) inputs

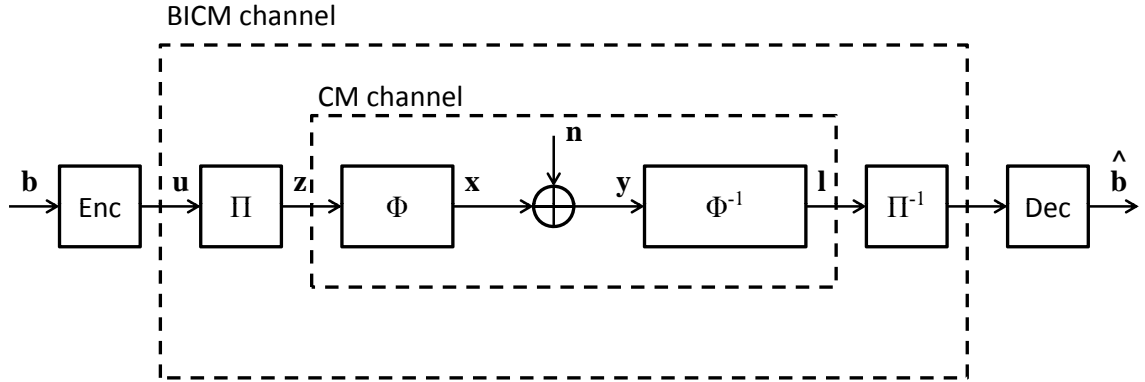


Figure 2.2: A BICM system structure which contains the CM system. The block ‘Enc’ is the binary channel encoder, ‘ $\Pi$ ’ is a bit-level interleaver and ‘ $\Phi$ ’ is a memoryless mapper. The blocks at the receiver side are the reverse processes.

interleaver has infinite depth and complete randomness. Then each bit is independent of the others and  $m$  bits are transmitted using independent symbols. The BICM capacity is defined as the capacity of the ‘BICM channel’ in Fig. 2.2. BICM system can be treated as an equivalent system with a set of  $m$  independent parallel channels connected to the encoder output by a random switch, which has the same function as the ideal interleaver. Each parallel channel corresponds to a bit position of symbols in  $\mathcal{X}$  and the BICM capacity can be treated as the capacity summed over the  $\mu$  parallel channels. Equation (2.11) is changed to (2.15) for generating BICM capacity when each bit level is demapped independently.

$$I(X; Y) = \sum_{i=1}^m E_{q,y} \left[ \log \sum_{x' \in \mathcal{X}_i^q} p(y|x') - \log \left( \sum_{x_k \in \mathcal{X}} p(y|x_k) p(x_k) \right) \right], \quad (2.15)$$

where  $\mathcal{X}_i^q$  defines the subset of  $\mathcal{X}$  whose corresponding  $i$ -th bit is  $q$  ( $q \in (0, 1)$ ). Unlike the CM capacity, the binary labeling affects the BICM capacity in (2.15) since each bit is treated independently. Fig. 2.3 shows the CM capacity and BICM capacity with 16-QAM modulation when each symbol is chosen with equal probability. Two different labeling rules are used to generate BICM capacity. The notation ‘capacity’ instead of SIR used in Fig. 2.3 with uniform input distribution is followed by the terminology in literature [34, 35]. Apparently by carefully choosing the labeling rule the capacity gap between CM and BICM capacity can be decreased. When  $x_k$  is assumed to be transmitted with identical probability the BICM capacity expression in (2.15) can be described as:

$$\hat{C} = \log_2(M) - \sum_{i=1}^m E_{q,y} \left[ \log \frac{\sum_{x' \in \mathcal{X}} p(y|x')}{\sum_{x' \in \mathcal{X}_i^{(q)}} p(y|x')} \right], \quad (2.16)$$

The expectation of equation (2.15) and (2.16) are both over the joint pdf  $p(q, y)$ .

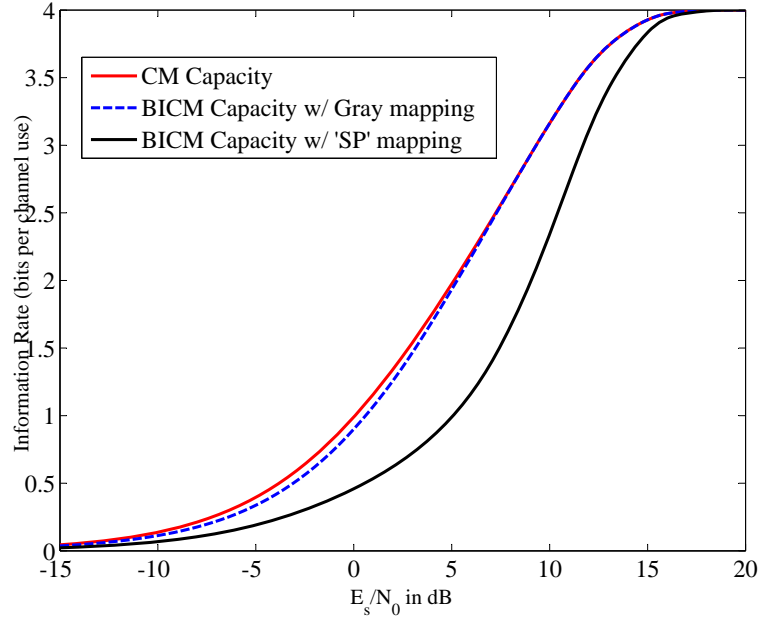


Figure 2.3: CM and BICM capacity for 16-QAM with uniform distribution, the BICM capacities are generated with two labeling rules

### 2.1.3 Capacity Calculation

The constrained capacity in equation (2.12) can be rewritten as,

$$\begin{aligned}
 C &= \max_{p(\mathbf{x})} \sum_{i=1}^M \int_{R^D} p(y|x_i) p(x_i) \log \left[ \frac{p(y|x_i)}{\sum_{k=0}^{M-1} p(x_k) p(y|x_k)} \right] dy \\
 &= \max_{p(\mathbf{x})} \sum_{i=1}^M \int_{R^D} p(y|x_i) p(x_i) \log [p(y|x_i)] dy \\
 &\quad - \max_{p(\mathbf{x})} \sum_{i=1}^M \int_{R^D} p(y|x_i) p(x_i) \log \left[ \sum_{k=0}^{M-1} p(x_k) p(y|x_k) \right] dy, \tag{2.17}
 \end{aligned}$$

where  $p(\mathbf{x}) = [p(x_0), p(x_1), \dots, p(x_{M-1})]$  is the joint pmf of the vector  $\mathbf{x}$ . For AWGN channel with zero mean and variance  $N_0/2$  noise, the channel transition probability is represented by:

$$p(y|x_i) = \frac{1}{(\sqrt{\pi N_0})^D} e^{-(y-x_i)^2/N_0}. \tag{2.18}$$

where  $D$  is the number of dimensions for the input symbols. When assuming the input symbols are uniform distributed, i.e.,  $p(x_i) = 1/M$ . The capacity becomes the symmetric

information rate (SIR)  $\hat{C}$  and equation (2.17) can be simplified as:

$$\hat{C} = \log(M) - \frac{1}{M(\sqrt{\pi})^D} \sum_{i=0}^{M-1} \int_{R^D} \exp(-t^2) \log \left[ \sum_{k=0}^{M-1} \exp(-2t \cdot d_{ik} - d_{ik}^2) \right] dt \quad (2.19)$$

where  $d_{ik} = (x_i - x_k)/\sqrt{N_0}$ . The detailed steps to derive equation (2.19) can be found in [36].

A numerical algorithm for evaluating the integration in (2.19) is based on the Gauss-Hermite quadrature, which could be used to approximate the integrals of the following kind:

$$\int_{-\infty}^{\infty} e^{-z^2} f(z) dz.$$

For a one-dimensional integral:

$$\int_{-\infty}^{\infty} e^{-z^2} f(z) dz \approx \sum_{i=1}^I w_i f(z_i) \quad (2.20)$$

where the points  $z_i$  are the roots of the *Hermite polynomials* and the coefficients  $w_i$  are the associated weights. The parameter  $I$  is the number of sample points to use for the approximation. Larger value of  $I$  lead to more accurate approximation results and lead to more computational complexity. For integration over  $D > 1$  the approximation is:

$$\int_{-\infty}^{\infty} e^{-\|\mathbf{z}\|^2} f(\mathbf{z}) d\mathbf{z} \approx \sum_{i_1=1}^I \dots \sum_{i_D=1}^I w_{i_1} w_{i_2} \dots w_{i_D} f(z_{i_1}, z_{i_2}, \dots, z_{i_D}) \quad (2.21)$$

The CM capacity in (2.19) can be generalized to a D-dimension  $\mathbf{t} = [t_1, \dots, t_D]$  and the SIR can be approximated by Gaussian Hermite quadrature as:

$$\begin{aligned} \hat{C} &= \log(M) - \frac{1}{M(\sqrt{\pi})^D} \sum_{i=0}^{M-1} \int_{R^D} \exp(-\mathbf{t}^2) \log \left[ \sum_{k=0}^{M-1} \exp(-2\mathbf{t} \cdot \mathbf{d}_{ik} - \mathbf{d}_{ik}^2) \right] d\mathbf{t} \\ &\approx \log(M) - \sum_{i_1=1}^I \dots \sum_{i_D=1}^I \prod_{d=1}^D w_{i_d} g_u^{\text{CM}}(\xi) \end{aligned} \quad (2.22)$$

where  $\xi = [\xi_{i_1}, \dots, \xi_{i_D}]$ , and the function  $g_u^{\text{CM}}(\xi)$  is:

$$g_u^{\text{CM}}(\xi) = \frac{1}{M(\sqrt{\pi})^D} \sum_{i=0}^{M-1} \log \left[ \sum_{k=0}^{M-1} \exp(-2\xi \cdot \mathbf{d}_{ik} - \mathbf{d}_{ik}^2) \right] \quad (2.23)$$

When the input distribution of  $x_i$  is not uniform, Gaussian Hermite quadrature still can be used based on equation (2.17):

$$I_{\text{CM}} = \sum_{i_1=1}^I \dots \sum_{i_D=1}^I \prod_{d=1}^D w_{i_d} g_p^{\text{CM}}(\xi) \quad (2.24)$$

$$(2.25)$$

where

$$g_p^{\text{CM}}(\xi) = -\frac{1}{(\sqrt{\pi})^D} \sum_{i=0}^{M-1} P_{\mathbf{X}}(\mathbf{x}_i) \log \left[ \sum_{k=0}^{M-1} P_{\mathbf{X}}(\mathbf{x}_k) \exp(-2\xi \cdot \mathbf{d}_{ik} - \mathbf{d}_{ik}^2) \right]$$

The BICM SIR could be numerically evaluated by Gaussian Hermite quadrature and the expression can be found in [37]. For the integration over multiple dimension  $D > 1$ , the Gaussian Hermite quadrature approximation has  $I^D$  terms for the part of summation. Therefore, it becomes impractical for large dimension  $D$ . For small values of  $D$ , i.e.,  $D = 1, 2$ , which are also the situations of practical interest, Gaussian Hermite quadrature approximation provides an alternative method to calculate the channel capacity. The capacity results computed in this dissertation are obtained with this approach with the number of sample points  $I$  set to be 15. The value generated by Gaussian Hermite quadrature approximation and Monte-Carlo simulation are almost identical, which can be observed from the capacity of 32-APSK modulation in Fig. 2.4.

The time required to compute the approximation is much shorter than that of simulation averaging. It took about several hours to generate the curve in Fig. 2.4 with simulation using a step size of 0.2 dB and 50 000 trials. However it took less than 5 minutes for Gaussian Hermite quadrature approximation to generate the same results. Based on the accuracy and speed advantage, Gaussian-Hermite quadrature approximation is chosen to calculate capacity or SIR values in the following discussion.

## 2.2 Constellation Shaping

Based on the previous description of channel capacity, the mutual information is not necessarily maximized by a uniform input distribution. The difference between the Shannon



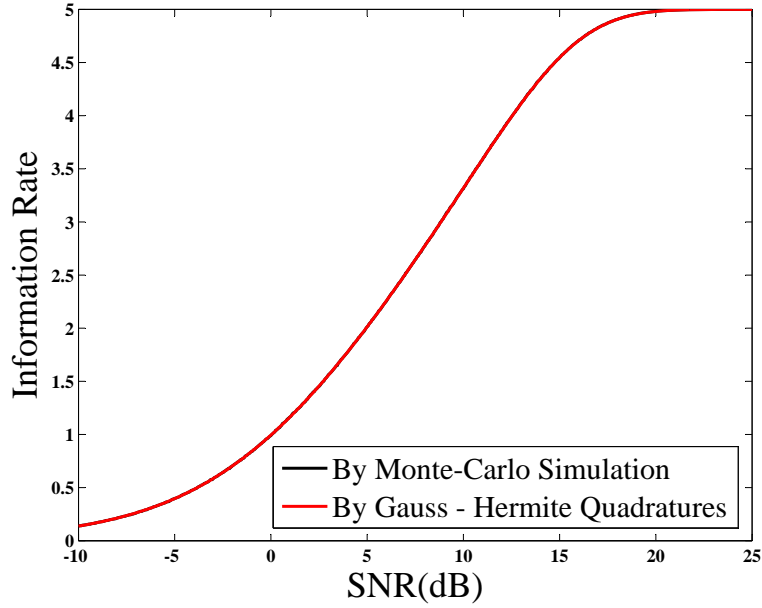


Figure 2.4: Information rate results of uniform 32-APSK modulation generated by two methodologies

capacity and SIR is known as the *shaping gain*, which is not realized in typical implementations involving uniformly distributed input signals. A strategy named *constellation shaping* that minimizes the gap between Shannon capacity and CM capacity generated by the uniform distributed symbols is introduced below.

The idea of constellation shaping is to transmit signals with small norm or energy at higher probability and transmit signals with large norm or energy less frequently. The nonuniform signals reduce the average bit rate of the transmitter, but the saved energy by selecting smaller norm signals more frequently may compensate for the rate loss [38]. Theoretically, constellation points would be selected according to a continuous Gaussian distribution at every dimension to achieve the maximum shaping gain. A smaller gain could be achieved in practical case when finite constellation is used.

One example of using constellation shaping is in the framework of multilevel codes [39], however, the increased complexity, time delay caused by multistage decoding and the design rules make it less attractive. Another example is the trellis shaping by using prefix codes [40, 32]. In [41] and [42], Le Goff et al. proposed a shaping technique suitable for bit-interleaved systems typical of modern communication standards. The technique uses the short nonlinear

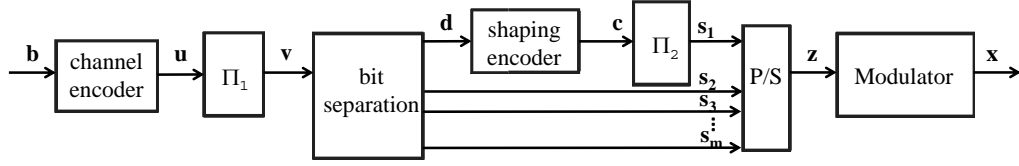


Figure 2.5: Transmitter structure.

shaping codes described in [43] to select from among a plurality of subconstellations. The constellation is partitioned into subconstellations such that the ones with lower average energy will be selected more frequently than those with higher energy. The details of this shaping technique is described below.

### 2.2.1 Shaping System Model

Inspired by the methodology proposed by Le Goff et al. in [41, 42], the constellation shaping strategy is extended to the LDPC-coded modulation system discussed below. A block diagram of the transmitter with constellation shaping is shown in Fig. 2.5. The input to the transmitter is information data with length- $k_c$ , which is fed into the channel encoder in the first place. The channel encoder produces a length- $n_c$  codeword  $\mathbf{u}$ , which is permuted by interleaver  $\Pi_1$  to produce the vector  $\mathbf{v}$ . A *bit separator* separates  $\mathbf{v}$  into  $m$  streams, where  $m = \log_2 M$  and  $M$  is the cardinality of modulation constellation, i.e., the number of symbol points in the constellation. The first bit stream, denoted  $\mathbf{d}$ , is passed through a shaping encoder to produce the shaping codeword  $\mathbf{c}$ , which is then permuted by interleaver  $\Pi_2$  to produce the vector  $\mathbf{s}_1$ . The purpose of the shaping encoder, as described below, is to produce a nonuniform binary output with a higher likelihood of 0's than 1's. There could be multiple streams instead of one single stream passing into the shaping encoder, the specific number is decided by the modulation and the desired probability distribution of the constellation symbols. For the reason of simplicity, only one streams is passed into the shaping encoder in Fig. 2.5 and we assume it is the first stream  $\mathbf{s}_1$  without loss of generality. The remaining streams at the output of the bit separator are denoted  $\{\mathbf{s}_2, \dots, \mathbf{s}_m\}$  and kept unchanged from the bit separation, they do not need to pass through the shaping encoder. The  $m$  streams  $\mathbf{s}_i$  are all of equal length and pass through a parallel-to-serial converter (P/S) and into the

modulator. Since each  $\mathbf{s}_i$  is of the same length and the shaping encoder has a rate less than unity, it follows that the stream pass into the shaping encoder is actually shorter than other streams outputted by the bit separator.

The P/S converter assures that each constellation symbol is selected using one bit from each  $\mathbf{s}_i$ . One bit position helps choose between the two subconstellations since only binary case is considered. Based on the fact of more zeros outputted by the shaping encoder, the sub-constellation with lower energy can be chosen more frequently by combining the proper constellation symbol labeling. The output of shaping encoder (e.g.,  $\mathbf{s}_1$  in the figure) is called *shaping bit* and it helps select symbols with nonequal probability. Normally, the lower-energy subconstellation is selected when the shaping bit is equal to 0, and the higher-energy subconstellation is selected when it is 1. The remaining bits, which are not shaped, are used to select from among the  $M/2$  symbols within the subconstellation. Since equal probability of 0s and 1s is kept in this bit position, symbols belonging to the same subconstellation are chosen with equal probability.

Note that there is just a single shaped bit passing to the shaping encoder in Fig. 2.5, which corresponds to partitioning the constellation into two subconstellations. More generally,  $g$  bits could be shaped, in which case the constellation is partitioned into  $2^g$  subconstellations [44]. To accommodate such a possibility, we use the shorthand  $\mathbf{s}_{1:g}$  to represent the first  $g$  streams, which have passed through the shaping encoder, and  $\mathbf{s}_{g+1:m}$  to represent the other  $m - g$  streams, which do not pass through the shaping encoder. The  $g$  streams may not be adjacent and the notation  $\mathbf{s}_{1:g}$  is used without loss of generality.

Prior to encoding, the shaping encoder's input  $\mathbf{d}$  is segmented into  $L$  short blocks based on the shaping code data bits  $k_s$  and length of  $\mathbf{d}$ , in other words, length of  $\mathbf{d}$  is the product of  $L$  and  $k_s$ . The encoder operates on each of block by mapping  $k_s$  bits onto a length- $n_s$  shaping codeword based on the mapping rules indicated by the shaping codeword table. The  $L$  code blocks that are generated by the encoder are then reassembled into a length- $Ln_s$  shaping codeword  $\mathbf{c}$ . Let  $p_0$  denote the probability that a particular bit in  $\mathbf{c}$  (or  $\mathbf{s}_1$ ) equals zero, and  $p_1$  be the probability that it equals one. Note  $p_0$  should be always greater than  $p_1$  according to the construction methodology of the shaping codeword table.

The shaping codeword table  $\mathcal{C}$  for shaping code  $k_s, n_s$  is a size  $2^{k_s} \times n_s$  matrix. Each

row of  $\mathcal{C}$  is a length- $n_s$  codeword. A unique length- $k_s$  binary input sequence is used to select a particular row from the matrix. The codeword table  $\mathcal{C}$  is initialized to contain the all-zeros codeword of length  $n_s$ , and codewords with higher weight are recursively added to  $\mathcal{C}$ . Suppose that  $\mathcal{C}$  contains all codewords of weight  $(w - 1)$  or lower but the number of rows in  $\mathcal{C}$  is still less than  $2^{k_s}$ . Then weight- $w$  codewords are drawn and added to  $\mathcal{C}$  until the number of distinct codewords is  $2^{k_s}$ . To assure each bit position in the codeword has approximately the same value of  $p_0$ , each new weight- $w$  codeword is selected with the goal of balancing the column weights of the codeword table  $\mathcal{C}$ .

As an example, consider the  $(n_s, k_s) = (5, 3)$  code. There are  $2^3 = 8$  codewords in codeword table  $\mathcal{C}$ . The number of binary codeword of weight two or less is:

$$\binom{5}{0} + \binom{5}{1} = 6.$$

In addition to these six codewords,  $\mathcal{C}$  will contain two more codeword of weight 2. There are  $\binom{5}{2} = 10$  possible weight-2 codewords to chose from when creating  $\mathcal{C}$ . For instance, if the last two codewords in  $\mathcal{C}$  are  $\{(00011), (00101)\}$ , then the column weights would be  $[1 \ 1 \ 2 \ 2 \ 3]$ . Alternatively, if they are  $\{(01100), (00011)\}$ , then the weights would be  $[1 \ 2 \ 2 \ 2 \ 2]$ . In the second design, the probability  $p_0$  of each bit position, which is equal to the column weight divided by 8 (the number of rows), is nearly constant. Thus, the second design is preferred over the first. This design policy is corroborated by the simulation results, which demonstrate improved performance when the column weights are carefully balanced.

The overall rate of the system  $R$  is the number of information bits per modulated symbol, and is related to the rates of the channel code and shaping codes rate by:

$$R = R_c [m + g(R_s - 1)] \quad (2.26)$$

where  $R_c$  is the rate of channel code and  $R_s$  is the rate of shaping code. When shaping is used,  $g > 0$  and  $R_s < 1$ , which implies that the rate of the channel code  $R_c$  used by a system with shaping must be higher than the  $R_c$  used without shaping if the overall system rate  $R$  is to remain fixed. In other words, the loss of the system rate  $R$  caused by the shaping code must be compensated by an appropriate increase in  $R_c$ .

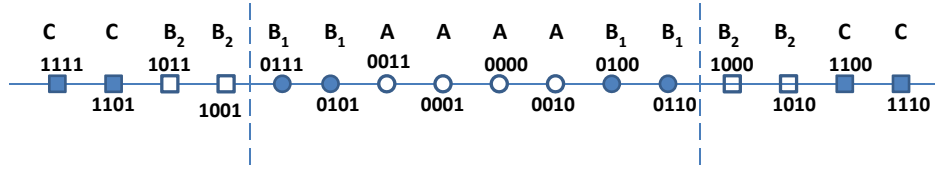


Figure 2.6: 16-PAM constellation.

## 2.2.2 Shaping Strategies

### Shaping for 16-PAM

Pulse-amplitude modulation (PAM) is used in some versions of the Ethernet communication standard and the more popular QAM modulation can be treated as two-dimensional PAM constellation. Fig. 2.6 shows the 16-PAM constellation and also the bit mapping rule. Two instances of shaped 16-PAM are considered with constellation symbols divided into 2 or 4 subconstellations. When there are two sub-constellations, only one shaping bit is needed, i.e.,  $g = 1$ , which is in the first position of the four-bit based on the constellation labeling rule of Fig. 2.6. One subconstellation set contains the eight middle symbols marked by circles, while the other set contains the remaining eight points marked by squares. The first set is selected with probability  $p_0$ , while the second set is selected with probability  $p_1$ . In the case of  $g = 2$ , the shaping bits are the first two bits of the labeling word. The signal set is partitioned into four groups which are indicated in Fig. 2.6 as  $A$ ,  $B_1$ ,  $B_2$ ,  $C$ . Set  $A$  is selected with probability  $p_0^2$ , set  $B_1$  and  $B_2$  are selected with probability  $p_0p_1$ , and  $C$  is selected with probability  $p_1^2$ . Since  $p_0 > p_1$  due to the shaping codeword construction methodology described in section 2.2.1, the symbols with the least energy in set  $A$  are selected more frequently than symbols in other groups, while the constellation symbols in set  $C$  are selected with the lowest probability.

### 2.2.3 Shaping for 16-APSK and 32-APSK

The APSK modulation was introduced in chapter 1 where it was described as being suitable for transmission over nonlinear channels. Consider the 16-APSK constellation shown

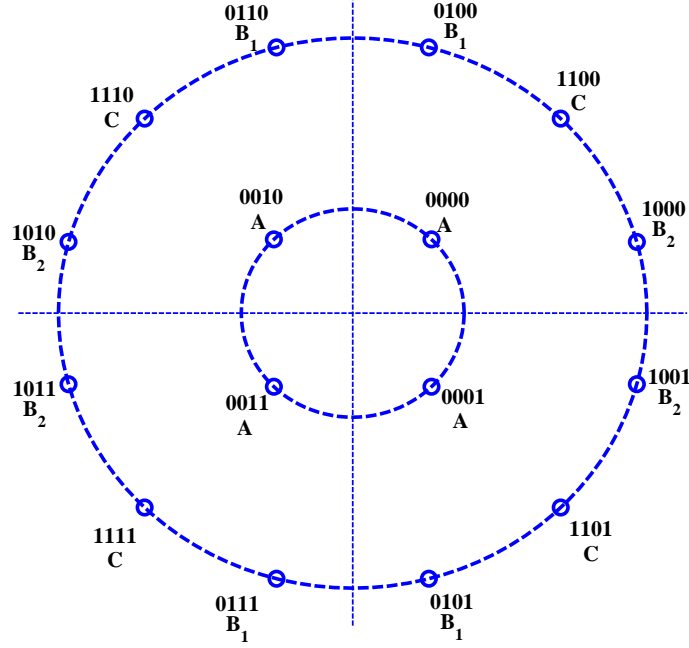


Figure 2.7: 16-APSK constellation.

in Fig. 2.7. The inner ring contains 4 points, while the outer ring contains 12 points. The bit mapping is as indicated on the figure. Note that this constellation is identical to the 16-APSK constellation in the DVB-S2 standard [19] and the mapping is identical except that the first two bits are complemented. The ratio of the radius of the outer ring to the radius of the inner ring is denoted  $\gamma$ , which according to the DVB-S2 standard must assume a value of 2.57, 2.60, 2.70, 2.75, 2.85, or 3.15.

Similar to the grouping of 16-PAM mentioned before, there are two different ways to divide 16-APSK constellation symbols into subgroups. In the case of  $g = 1$  shaping bit, the shaping bit is the first bit position and the signal constellation is partitioned into two sets. The first set contains points labeled  $A$  and  $B_1$  in Fig. 2.7 and is selected with the probability  $p_0$ , while the second set contains points labeled  $B_2$  and  $C_1$  and is selected with probability  $p_1$ . Note that each symbol inside the subconstellation is chosen with equal probability.

In the case that  $g = 2$ , the shaping bits are the first two bits of the labeling word. The signal partitions are the same as that of 16-PAM. Set  $A$  is selected with probability  $p_0^2$ , set  $B_1$  and  $B_2$  are each selected with probability  $p_0 p_1$ , and set  $C_1$  is selected with probability  $p_1^2$ .

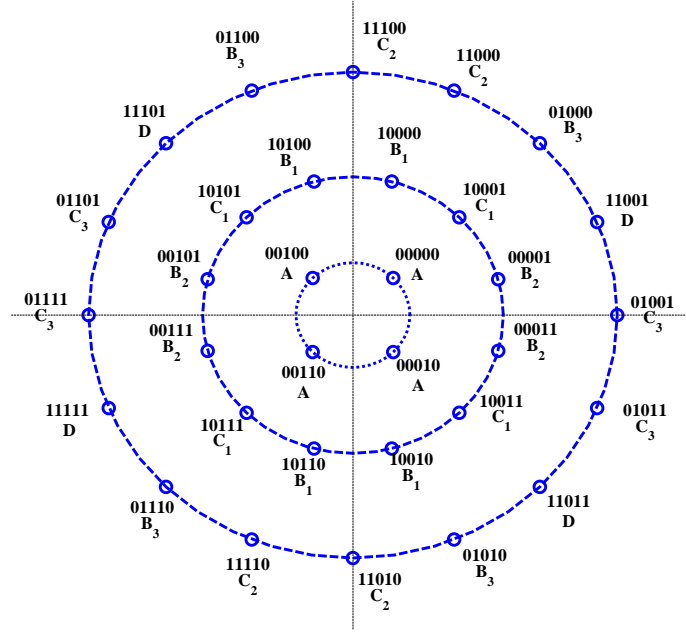


Figure 2.8: 32-APSK constellation.

A 32-APSK constellation is shown in Fig. 2.8 along with its bit mapping. The constellation consists of three concentric rings, with 4 points in the inner ring, 12 points in the middle ring, and 16 points in the outer ring. This constellation is identical to the 32-APSK constellation in the DVB-S2 standard [19] except that the first and last bits of the mapping word are complemented. The ratio of the radius of the middle ring to the radius of the inner ring is denoted  $\gamma_1$ , while the ratio of the radius of the outer ring to the radius of the inner ring is denoted  $\gamma_2$ . According to the standard, the pair  $\{\gamma_1, \gamma_2\}$  must be one of the following:  $\{2.53, 4.30\}$ ,  $\{2.54, 4.33\}$ ,  $\{2.64, 4.64\}$ ,  $\{2.72, 4.87\}$ , or  $\{2.84, 5.27\}$ .

The number of minimum energy signals is four, which is the size of the smallest partition. The number of shaping bits should be no more than three. In the case of  $g = 1$  shaping bit, the shaping bit is the second bit of the word. The constellation is divided into two partitions. The first partition, which is selected with probability  $p_0$ , contains the signals in the first two rings (i.e., sets  $A$ ,  $B_1$ ,  $B_2$ , and  $C_1$ ). The second partition, which is selected with probability  $p_1$ , contains the signals in the outer ring (i.e., sets  $B_3$ ,  $C_2$ ,  $C_3$ , and  $D$ ). With this scheme, signals in the inner two rings are more likely to be selected than signals in the outer ring.

When  $g = 2$ , the shaping bits are the second and last bits of the codeword. The signal

set is partitioned into four sets:  $\{A, B_1\}$ ,  $\{B_2, C_1\}$ ,  $\{B_3, C_2\}$ , and  $\{C_3, D\}$ . The partition  $\{A, B_1\}$  is selected with probability  $p_0^2$ , making it most likely to be selected. The least likely group to be selected is  $\{C_3, D\}$  with probability  $p_1^2$ . The other two partitions are each selected with probability  $p_0p_1$ .

When  $g = 3$ , the shaping bits are the first, second, and last bits of the word. The signal set is partitioned into the eight sets indicated in Fig. 2.8; i.e.,  $A$ ,  $B_1$ ,  $B_2$ ,  $B_3$ ,  $C_1$ ,  $C_2$ ,  $C_3$ , and  $D$ . Partition  $A$  is most likely and is selected with probability  $p_0^3$ . The  $B_k$  partitions are selected with probability  $p_0^2p_1$ , while the  $C_k$  partitions are selected with probability  $p_0p_1^2$ . Finally, the  $D$  partition is selected with probability  $p_1^3$ , which makes it least likely to be selected.

## 2.3 Parameter Optimization

With the shaping techniques described in the previous section, energy may be conserved by using larger values of  $p_0$ . However, larger values of  $p_0$  generally require lower shaping code rate  $R_s$ . To maintain a fixed overall rate  $R$  from equation (2.26), using a lower  $R_s$  requires a larger value of  $R_c$ , which weakens the effectiveness of the error control code. Clearly there is a tradeoff between  $R_s$  and  $R_c$ .

Rather than allowing arbitrary  $p(x)$ , the system with the proposed shaping techniques is considered, thus  $p(x)$  is under the constraint of number of shaping bit and chosen shaping code. For a particular constellation  $\mathcal{X}$ , number of shaping bits  $g$ , and  $(n_s, k_s)$  shaping code, we numerically evaluate the information rate  $I(X; Y)$  by using Gauss-Hermite quadratures approximation from equation (2.25).

The parameter optimization is started by evaluating the information rates under the assumption that the input symbols have a uniform distribution. The generated SIR value is served as a basic benchmark for the further comparison. The SIR of one dimensional constellation 16-PAM is shown in Fig. 2.9 as a function of  $\mathcal{E}_s/N_0$ . For the calculation of APSK modulation, the radius ratios  $\gamma$  can affect the information rate value thus SIR with different  $\gamma$  is evaluated at a single  $\mathcal{E}_s/N_0$  point. The possible radius ratio choices are constrained as those defined in DVB-S2 standard for the reason of simplicity. For  $M = 16$ ,



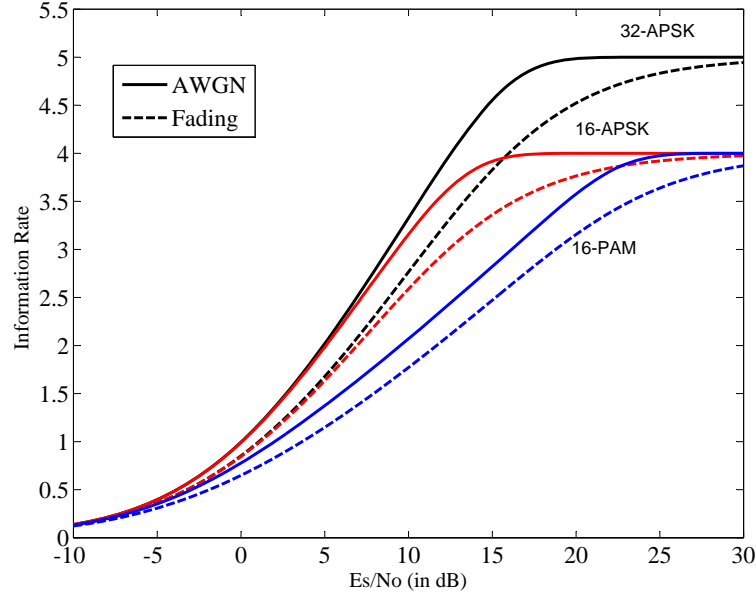


Figure 2.9: Information rate (in bits per channel use) of 16-APSK, 16-PAM and 32-APSK with uniform input distributions over AWGN and ergodic Rayleigh fading channels. The rates for APSK modulation are maximized over the permissible ring radius ratios.

the information rate is evaluated as a function of  $\mathcal{E}_s/N_0$  for each of the six values of  $\gamma$  specified in the DVB-S2 standard. For each value of  $\mathcal{E}_s/N_0$ , we note the maximum value of the information rate over all choices of  $\gamma$ . Similarly, for  $M = 32$ , we evaluated the information rate for each of the five values of  $\{\gamma_1, \gamma_2\}$  specified in the DVB-S2 standard, and we noted the maximum information rate at each  $\mathcal{E}_s/N_0$ . The symmetric information rate of APSK modulation, maximized over the permissible ring radius ratios, is shown in Fig. 2.9.

The information rate over ergodic Rayleigh fading channels is also plotted in Fig. 2.9. The SIR in fading is found by taking the expected value of the information rate with respect to the channel SNR. More specifically, let  $I(\beta)$  be the information rate as a function of SNR  $\beta$ . It follows that in fading, the information rate is equal to

$$E[I(\beta)] = \int p(\beta)I(\beta)d\beta = \int_0^\infty \frac{1}{\Gamma} \exp\left\{-\frac{\beta}{\Gamma}\right\} I(\beta)d\beta \quad (2.27)$$

where  $p(\beta)$  is the pdf of an exponential random variable with mean  $\Gamma = \mathcal{E}_s/N_0$ . The average is found through numerical integration.

The next step is to compute information rates when constellation shaping is used. For each constellation mentioned in section 2.2.2, all valid shaping codes with  $n_s \leq 20$  and

$k_s \leq 10$  are considered. Larger values are ignored due to their high decoding complexity (which is exponential in  $k_c$ ). Each shaping code corresponds to a value of  $p_0$ . For a given number of shaping bit  $g$  and a given  $p_0$ , the constellation pmf  $p(\mathbf{x})$  can be determined based on the shaping strategy given by section 2.2. The information rate can thus be computed through numerical calculation. The total number of distinct  $p_0$  that we considered is 121, and this exercise was repeated for each permissible value of  $\gamma$  or  $\{\gamma_1, \gamma_2\}$  when using APSK modulation. For each constellation with different number of shaping bit  $g$ , we determine the maximum information rate at each value of  $\mathcal{E}_s/N_0$ . The 16-PAM constellation is easier because no radius ratios need to be considered, while the rate is maximized over both the permissible  $p_0$  and ring radius ratios for 16APSK and 32APSK. We process this optimizing strategy for AWGN channel as well as the ergodic Rayleigh fading channel with three different modulations.

Based on the search results, shaping gain up to 0.78 dB is achieved with the proposed shaping strategies. Fig. 2.10 shows the information rates with constellation shaping for 16-APSK, 16-PAM and 32-APSK in AWGN and Fading channel. The information rate at each  $\mathcal{E}_s/N_0$  is maximized over the choice of all 121 shaping codes and over the radius ratios for the APSK constellation. The results of 16-APSK show the benefit of using two shaping bits over just one shaping bit. One shaping bit seems to be working better for 16-PAM based on the generated results. The shaping gain is almost the same when using 1 shaping bit and 2 shaping bits so the former strategy is preferred to reduce additional system complexity. The results for 32-APSK show that while 3 shaping bits is better than using just 1 shaping bit, the gain of using 2 shaping bits is actually less than the gain of 1 shaping bit. For this reason, it is not advisable to use 2 shaping bits in the 32-APSK case. Furthermore, the incremental gain of using 3 shaping bits over 1 shaping bit is quite marginal, and it is recommended that 32-APSK systems use just 1 shaping bit. This result has an intuitive justification: with one shaping bit, the system will simply choose from the outer ring or the inner two rings. The shaping gain could be visualized by measuring the curve distance between the shaped system and uniform system, in other words, it is the SNR difference when both shaping system and uniform system has the same information rate. Fig. 2.11 shows the shaping gain when different shaping bits are used for three modulations over AWGN and Fading channel.

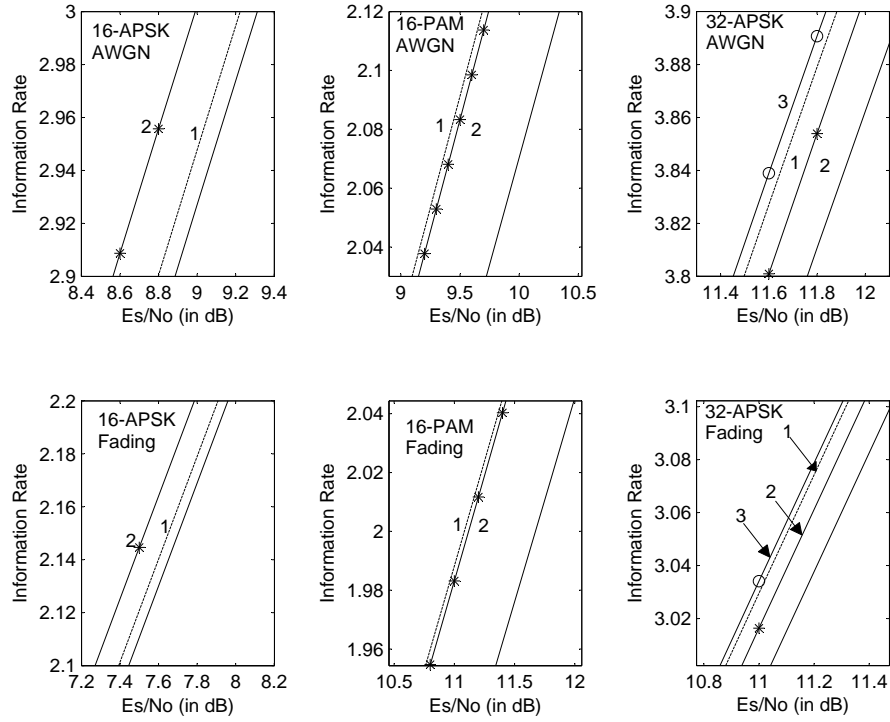


Figure 2.10: Information rate of nonuniform APSK and 16-PAM. The number of shaping bit used is indicated. The solid black line on the right of each subfigure corresponds to a uniform input distribution. The results of AWGN Channel are shown in the first row, the results of ergodic Rayleigh fading channel are shown in the second row. Subfigures in leftmost column: 16-APSK with 1 or 2 shaping bits. Subfigures in the middle column: 16-PAM with 1 or 2 shaping bits. Subfigures in last column: 32-APSK with 1, 2, or 3 shaping bits.

The results from Fig. 2.10 and 2.11 are tabulated for AWGN in Table 2.1 and for ergodic Rayleigh fading in Table 2.2. For each constellation and possible  $g$ , the tables show the rate  $R$  for which shaping has the highest gain, along with the permissible values of  $p_0$ . Radius ratios  $\gamma$  that achieve the gain are also shown when APSK modulation is used. The value of  $\mathcal{E}_b/N_0$  for which the shaped constellation achieves rate  $R$  is listed, where  $\mathcal{E}_b/N_0 = (1/R)(\mathcal{E}_s/N_0)$  is the SNR per information bit.

Fig. 2.12 shows the value of  $p_0$  that maximizes the information rate for 16-PAM and 32-APSK with  $g = 1$  shaping bits, the results for 16-APSK with  $g = 2$  shaping bit are shown in the same figure. The general trend is that lower  $p_0$  is optimal for higher SNR, and fading channels may require smaller value of  $p_0$  than AWGN channels do.

Table 2.1: Minimum required  $\mathcal{E}_b/N_0$  in AWGN for 16-PAM and M-APSK with  $g$  shaping bits. The optimizing  $p_0$  and  $\gamma$  are shown.

Mod	$g$	$R$	$\mathcal{E}_b/N_0$	$gain$	$p_0$	$\gamma$
16-PAM	1	3.03	10.78 dB	0.771 dB	0.7287	N/A
	2	3.09	11.1 dB	0.768 dB	0.6875	N/A
16-APSK	1	3.09	4.714 dB	0.091 dB	0.623	2.70
	2	2.95	4.077 dB	0.322 dB	0.688	2.57
32-APSK	1	3.88	5.915 dB	0.265 dB	0.716	{2.64,4.64}
	2	4.06	6.517 dB	0.175 dB	0.623	{2.53,4.30}
	3	3.89	5.898 dB	0.310 dB	0.656	{2.53,4.30}

Table 2.2: Minimum required  $\mathcal{E}_b/N_0$  in ergodic Rayleigh fading for 16-PAM and M-APSK with  $g$  shaping bits. The optimizing  $p_0$  and  $\gamma$  are shown.

Mod	$g$	$R$	$\mathcal{E}_b/N_0$	$gain$	$p_0$	$\gamma$
16-PAM	1	2.43	10.24 dB	0.631 dB	0.7652	N/A
	2	2.48	10.46 dB	0.6095 dB	0.6799	N/A
16-APSK	1	2.01	3.897 dB	0.052 dB	0.623	2.85
	2	2.14	4.172 dB	0.175 dB	0.656	2.57
32-APSK	1	3.03	6.188 dB	0.164 dB	0.688	{2.64,4.64}
	2	3.13	6.551 dB	0.106 dB	0.623	{2.54,4.33}
	3	3.03	6.167 dB	0.184 dB	0.623	{2.53,4.30}

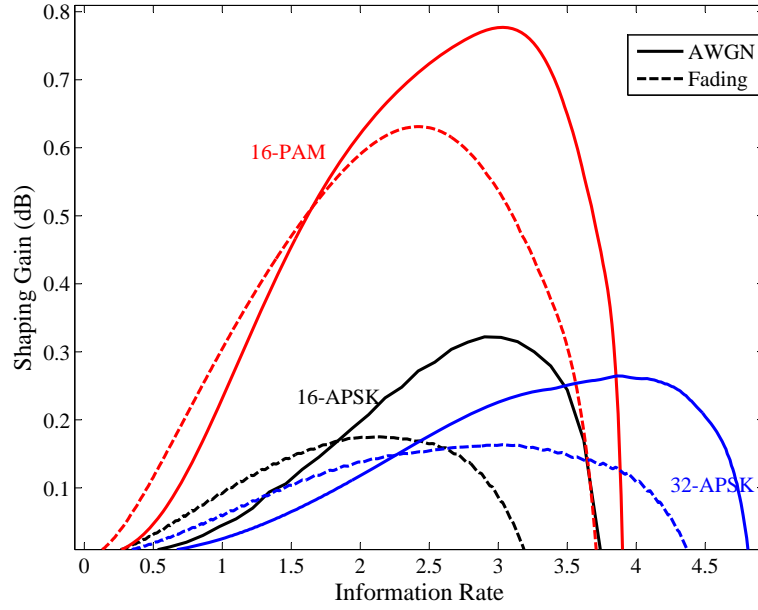


Figure 2.11: Shaping gain of 16-PAM, 16-APSK and 32-APSK over AWGN and ergodic Rayleigh fading channel. For 16-PAM and 32-APSK, 1 shaping bits are used. For 16-APSK, 2 shaping bits are used

## 2.4 PAPR Consideration

The peak-to-average power ratio (PAPR) is a consideration when nonlinear power amplifiers need be used. The PAPR is defined as,

$$\text{PAPR} = \frac{\max_{x \in \mathcal{X}} \{|x|^2\}}{E[|x|^2]} = \frac{\max_{x \in \mathcal{X}} \{|x|^2\}}{\sum_{x \in \mathcal{X}} p(x) |x|^2} \quad (2.28)$$

Because the goal of shaping is to minimize average energy, it follows that shaping will increase the PAPR for a particular constellation  $\mathcal{X}$ . This behavior can be seen in Fig. 2.13, which shows the relationship between PAPR and  $p_0$  for the shaping strategies considered in this paper.

At first glance, it would seem that shaping will dramatically increase the PAPR. This is true if we were to use the same  $\mathcal{X}$  both with and without shaping. However, when we performed our optimization, we let the ratio of ring radii vary among their permissible values. When we did this, we found that the optimal ratio of ring radii were always smaller with shaping than without. Furthermore, the PAPR is smaller for smaller ring radii ratios. Thus,

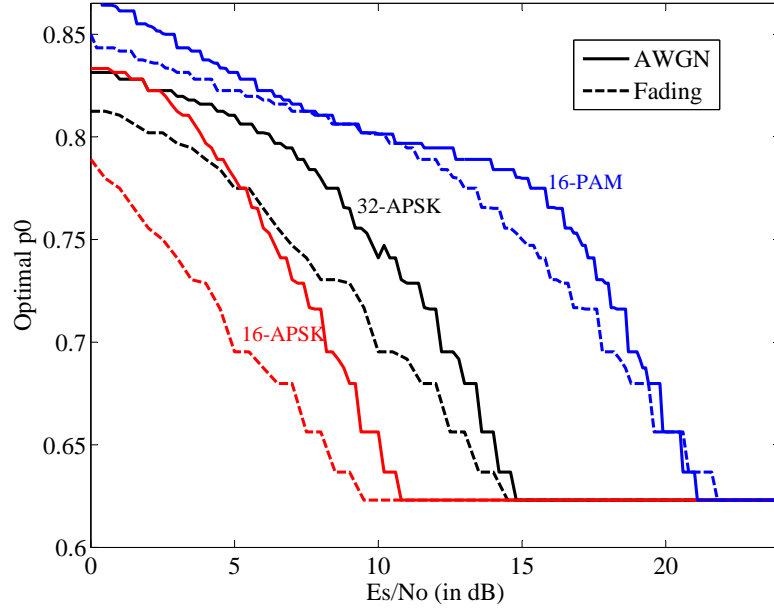


Figure 2.12: The value of  $p_0$  that maximizes the information rate for nonuniform APSK in AWGN and ergodic Rayleigh fading. For 16-APSK, 2 shaping bits are used, while for 16-PAM and 32-APSK, 1 shaping bit is used.

in achieving the balance between the ratio of ring radii and the value of  $p_0$ , the PAPR increase is not significant. For example, in the case of 16-APSK with 2 shaping bits, we found that the PAPR with shaping is 1.98 dB, while for the uniform case, which was optimized at the largest value of  $\gamma$ , was 1.11 dB, and thus only 0.86 dB additional PAPR is required when shaping.

## 2.5 Summary

This chapter gives an introduction to channel capacity, which is the mutual information between the channel input and output maximized over all possible input distributions. The capacity under modulation constraints is described, and the symmetric information rate is discussed as the mutual information achieved at the uniform input distribution. BICM capacity is then introduced, followed by the methodologies for numerically computing the mutual information. The second part of this chapter introduces the strategy of constellation shaping. The block diagram of transmitter including the shaping encoder is described. A

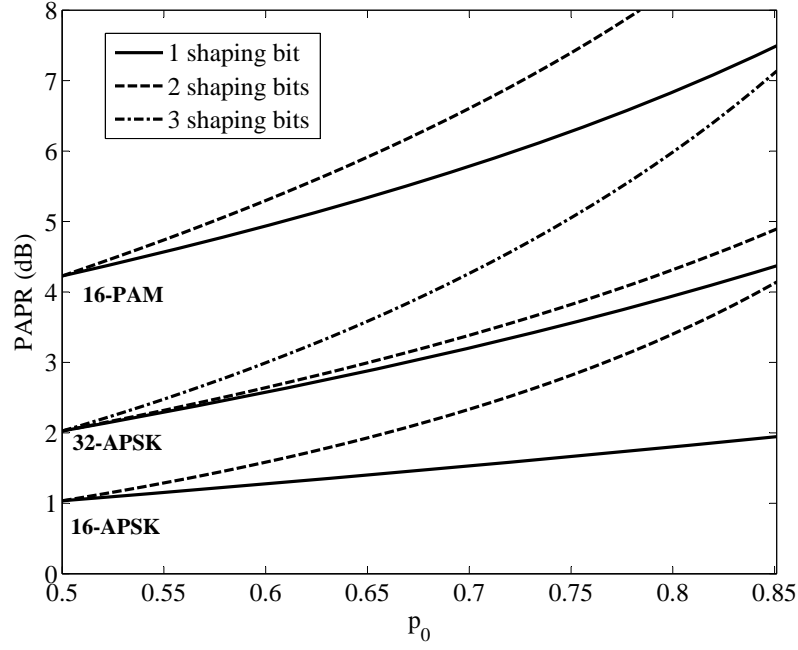


Figure 2.13: Peak-to-average power ratio (PAPR) of nonuniform 16-PAM, 32-APSK and 16-APSK as a function of  $p_0$  and the number of shaping bits. For 16-APSK,  $\gamma = 2.57$ , and for 32-APSK,  $\gamma = \{2.64, 4.64\}$ .

non-linear shaping code helps select the constellation symbols with non-uniform probability. Information rates are computed for a variety of modulation and shaping code parameters, in an effort to optimize these parameters. The jointly optimized results and PAPR comparison are shown at the end of this chapter. Details of system implementation with shaping encoder and decoder, along with the performance of both the shaping and uniform system are provided in the following chapter.

## Chapter 3

# Constellation Shaping Implementation

This chapter describes the implementation of the systems with constellation shaping strategy. The uniform system can also be evaluated by eliminating the components of shaping encoder and decoder. Both BICM-ID and BICM system are used as the coded modulation model in the following system implementation schemes. The BICM system was first appeared in 1992 as an alternative of *trellis coded modulation* over Rayleigh fading channel [45]. At the transmitter side, the bit-level interleaver separates the binary coding from the modulation, and the reliability information of the coded bits is propagated from the demodulator to the decoder through the corresponding de-interleaver at the receiver side. Caire et al. provided a comprehensive analysis of BICM to show its benefits [34]. Since all the operations in BICM is bit-wised, the coding and modulation could be chosen independently for the particular channel conditions. BICM is widely used in current wireless communications standards, including IEEE 802.11a/g, IEEE 802.16, and DVB-S2 [46]. Iterative demodulation for BICM was proposed by Li and Ritcey and ten Brink [47, 48, 49], the results indicate the significant performance gains compared with non-iterative BICM decoding. The performance comparison between BICM and BICM-ID, shaping and uniform systems are discussed in this chapter.

Turbo and LDPC codes are used in the following system implementation as the channel code options. The 16-PAM and 32-APSK are chosen to be the modulation schemes. Simulation results show that BICM-ID system performs better than that of BICM model by passing the extrinsic information generated by the decoder back into the demodulator. Besides, con-



stellation shaping strategy provides the capability to improve the error performance under the same signal to noise ratio as the uniform systems. The transmitter part including the shaping encoder is introduced in the beginning of this chapter. Then particular attention is paid to describe the components of the receiver, including the demodulator, shaping decoder and the channel code decoder. The final part of this chapter presents the numerical simulation results and the corresponding system parameters. In the following discussion, bold lowercase letters denotes the vectors, e.g.  $\mathbf{c}$ , bold uppercase will be used for matrices, like the LDPC codes' parity check matrix  $\mathbf{H}$ .

### 3.1 Transmitter Structure

Followed by the description of constellation shaping strategy in the previous chapter, this section gives a simple review of the transmitter with constellation shaping. And then the actual parameters used in the system implementation for numerical simulation are introduced. Only the shaping strategy with one shaping bit is considered in the actual implementation to strike a balance between the performance and system complexity.

As shown in Fig. 2.5, a binary vector of message bits  $\mathbf{b}$  is passed through a rate  $R_c = k_c/n_c$  binary channel encoder to produce a codeword  $\mathbf{u}$ . The codeword is then permuted randomly by a bit-based interleaver. The output  $\mathbf{v}$  of the interleaver is partially encoded by the rate  $R_s = k_s/n_s$  shaping encoder to generate the sequence  $\mathbf{z}$ . When the shaping code has unity code rate, i.e.,  $R_s = 1$ , then no constellation shaping is involved. The channel code can be either turbo or LDPC code in the system. Uniform constellation is considered first, and then the system with constellation shaping is implemented to compare the performance with the uniform system under the same spectral efficiency.

The bit separation in Fig. 2.5 is necessary only when constellation shaping is used. It divides the output vector  $\mathbf{v}$  of the interleaver into  $m = \log_2(M)$  streams  $\mathbf{s}_i$  and  $g$  out of  $m$  streams are passed to the shaping encoder. Each  $m$  streams should have equal length so that the modulator chooses one bit from each bit stream and maps  $m$  bits into one of  $M = 2^m$  constellation symbols. The bit separation is asymmetric because the shaping code rate is always less than one, i.e.,  $R_s < 1$ . According to the optimized parameter results in section

2.3, the modulation which are separated into two sub-constellations could yield shaping gain without increasing the system complexity dramatically. In the following system schemes,  $g = 1$  is used and only 1 out of  $m$  bit streams is passed into shaping encoder, which is noted by mentioning only 1 *shaping bit* is used. Recall the construction methodology of shaping codeword in section 2.2, the output of shaping encoder contains more zeros than ones, which means more lower energy symbols are transmitted through the noisy channel.

The overall rate of the system is the number of information bits per modulated symbol, and can be expressed by:

$$R = R_c [m + g(R_s - 1)]. \quad (3.1)$$

where  $R_c$  is the rate of channel code and  $R_s = k_s/n_s$  is the rate of shaping code. When constellation shaping is used,  $g > 0$  and  $R_s < 1$ , which indicates that for a fixed system rate  $R$ , the rate of the channel code  $R_c$  with shaping must be higher than the rate of the channel code when shaping is not used, i.e., system with uniform constellation.

### 3.1.1 Constellation Shaping with 16-PAM

When 16-PAM is used,  $M = 16$  and  $g = 1$  for the shaping system, the shaping bit is the first bit position according to the labeling of 16-PAM constellation points in Fig. 2.6. There are two sub-constellations grouped by the different value of the first bit in the labeling. The subconstellation with lower energy contains the 8 symbols close to the origin and each of them is chosen with probability  $p_0/8$ . The other subconstellation includes the rest 8 symbol points far away from the origin, each of which has larger energy than the symbols in the lower energy subconstellation and are chosen with probability  $p_1/8$ . Both turbo and LDPC codes are tested in uniform and shaping system over AWGN channel. The turbo code is the type specified in the UMTS standard [50] and the adopted LDPC codes are defined in DVB-S2 standard [19]. The parameters used for simulations are listed in section 3.3.

### 3.1.2 Constellation Shaping with 32-APSK

The 32-APSK constellation in Fig. 3.1 follows the DVB-S2 standard and consists of three concentric rings. When there are two equal sized sub-constellations ( $g = 1$ ), the

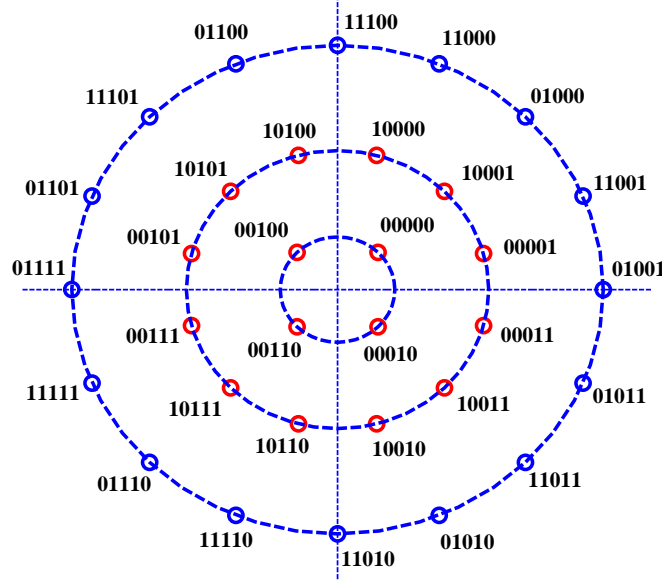


Figure 3.1: 32-APSK constellation.

shaping bit is the second bit of the word based on the labeling rule shown in Fig. 3.1. The first partition, which is selected with probability  $p_0$ , contains the signals in the two inner rings, i.e. constellation symbols noted by red circle. The second partition, which is selected with probability  $p_1$ , contains the signals in the outer ring. Due to the larger number of 0's outputted by shaping encoder, signals in the first partition are more likely to be selected than signals in second partition.

LDPC codes for a fixed set of eleven different rates from  $R_c = 1/4$  to  $R_c = 9/10$  are defined by DVB-S2 standard. According to equation (3.1), the possible choice of shaping code rates  $R_s$  is limited if the overall rate  $R$  is fixed. For example, the results in Table 2.1 indicates that the maximum performance gain for 32-APSK constellation with 1 shaping bit is achieved when the system rate  $R = 3.88$  over the AWGN channel, however, we could not find the LDPC code defined in DVB-S2 to achieve that rate when 32-APSK modulation is used. If the spectral efficiency  $R = 4$  is considered as an alternative, the shaping system should choose the combination with LDPC code rate  $R_c = 5/6$  and shaping code rate  $R_s = 4/5$ . But the  $p_0$  value of shaping code  $n_s = 5, k_s = 4$  may not be optimum based on the information rate analysis. The actual selected parameters may not be the same as those

shown in section 2.3, and the chosen ones are determined by striking the balance among different aspects.

## 3.2 Receiver Implementation

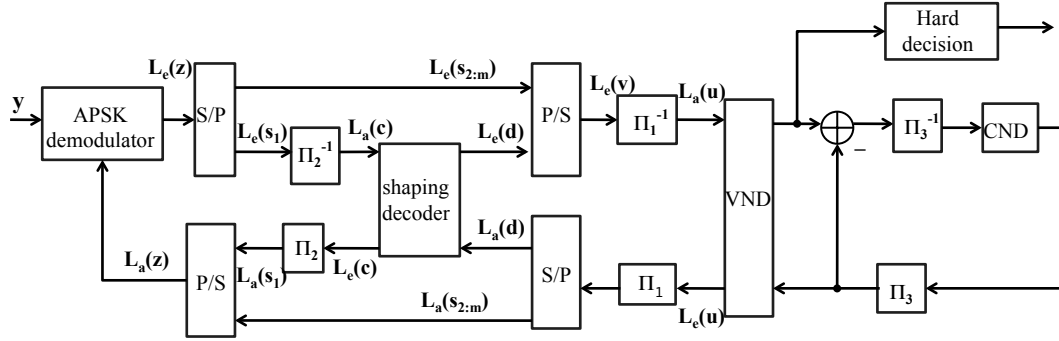


Figure 3.2: Receiver structure.

The receiver structure shown in Fig. 3.2 is the so-called BICM-ID model and the feedback information is used to improve the estimation produced by the demodulator. The BICM model without the feedback loop can be also referred to this figure. The structure of APSK demodulator and of LDPC decoder are shown in Fig. 3.2, the idea is the same when using PAM demodulator and turbo decoder. Throughout this whole section, all information exchanged within the receiver is assumed in the form of *log-likelihood ratios* (LLRs). The notation  $L_a$  denotes an LLR input to a module, which may possibly be fed back from another module; while  $L_e$  denotes the output LLRs of a module. The received signals  $\mathbf{y}$ , which are corrupted by the channel noise, are passed into the APSK demodulator. The LLRs generated by the APSK demodulator are divided into  $m$  streams  $L_e(s_i)$  by the serial-to-parallel operation (denoted by S/P in Fig. 3.2).  $L_e(s_{2:m})$  is used to represent the LLRs of streams  $\{\mathbf{s}_2, \dots, \mathbf{s}_m\}$ . Clearly there is no bit separation or bit combination in the system without constellation shaping, the output of the demodulator will pass into the channel decoder after a single de-interleaver.

The shaping system with both constellations (16-PAM and 32-APSK) use one shaping bit, which corresponds to one out of  $m$  streams pass into the shaping encoder. For the simplicity reason, the shaping bit lies in the *first* position in the receiver block diagram of

Fig. 3.2, even though the second bit position is the actual shaping bit when the modulation scheme is 32-APSK. The first stream containing the LLRs about the shaping codeword  $L_e(\mathbf{s}_1)$  is de-permuted to produce  $L_a(\mathbf{c})$  and fed into the shaping decoder along with the *a priori* information from the decoder at previous iteration. Initially, the *a priori* input to the shaping decoder  $L_a(\mathbf{d})$  is set to be all zeros. A parallel to serial converter combines the output of the shaping decoder  $L_e(\mathbf{d})$  with the LLRs of the other streams  $L_e(\mathbf{s}_{2:m})$  to produce a vector of combined LLRs  $L_e(\mathbf{v})$ . The vector  $L_e(\mathbf{v})$  is de-permuted by  $\Pi_1^{-1}$ , and the resulting vector  $L_a(\mathbf{u})$  is introduced as the input to the LDPC decoder.

When BICM-ID model is used in the receiver, the output  $L_e(\mathbf{u})$  of the channel decoder is fed back to the shaping decoder and demodulator, also called the *demapper*, to be used as extrinsic information during the next iteration. A second S/P converter sends the LLRs of the shaped bits into the shaping decoder as input  $L_a(\mathbf{d})$ , where after the initial iteration it is used as the *a priori* input. Another P/S converter, which is identical to the one in Fig. 2.5, combines the interleaved output of the shaping decoder with the LLRs of the unshaped bits to create the vector  $L_a(\mathbf{z})$ . The output of the P/S converter is used as the *a priori* input to the demapper after the initial iteration. Prior to the first iteration,  $L_a(\mathbf{z})$  is initialized according to the average bit probabilities for each bit position, as described in details below.

During each receiver iteration, four information exchange operations are performed: (1) between the demapper and channel decoder, (2) between the demapper and shaping decoder, (3) between the channel decoder and shaping decoder, and (4) inside the channel decoder. Notice that only one iteration of channel decoding is executed per iteration of demapping and shaping decoding. In fact, the shaping decoder is the only additional processor required for this system compared to the BICM-ID system in [51]. Although the inclusion of the shaping decoder increases the complexity of the receiver, the additional complexity is moderate provided that the reasonably small length of the shaping code.

The receiver consists of three important modules, which are described respectively in the following subsections about the demodulator, shaping decoder, and channel decoder. The demodulator and shaping decoder are the same when using different types of channel codes. The decoder of both turbo and LDPC code could be treated as the combination of two sub-decoders; two convolutional decoders are inside the turbo decoder and *variable node*

decoder (VND) connected with *check node decoder* (CND) inside LDPC decoder. Because information is exchanged among all components, the notion of an “iteration” may be ambiguous. In the following discussion, one full iteration is specifically meant to include an iteration of the shaping decoder, an iteration of the channel decoder, and for the case of BICM-ID, one iteration of symbol demapping. For the BICM system, the symbol demapper is only executed once, and information is not fed back to the demapper.

### 3.2.1 The Demodulator

The demodulator is implemented on a symbol by symbol basis. For ease of exposition, we drop the dependence on the symbol interval in this subsection, so that symbols may be expressed without subscripts. During a particular symbol interval, the demodulator computes the LLRs  $L_e(\mathbf{z})$  of the  $m$  code bits associated with the transmitted symbol  $x \in \mathcal{X}$ , where  $\mathcal{X}$  is the signal constellation. The inputs to the demodulator are the received discrete-time signal  $y$ , which is produced by a matched-filter front end, as well as the set of  $m$  *a priori* LLRs  $L_a(\mathbf{z})$  generated by the shaping and LDPC decoders during the previous iteration. Prior to the first iteration,  $L_a(\mathbf{z})$  is initialized to  $L_a(\mathbf{s}_1) = \log(\frac{1-p_0}{p_0})$  for the shaped bit and  $L_a(\mathbf{s}_{2:m}) = 0$  for the unshaped bits. The first stream  $\mathbf{s}_1$  corresponds to the shaping codewords, it follows that  $p_0 > 1/2$  and  $L_a(\mathbf{s}_1)$  will initially be negative.

Let the function  $z_k(x)$  return the  $k^{th}$  bit labeling symbol  $x$ . Using the *maximum a posteriori probability* (MAP) demodulator described in [42] and [52], the a posteriori probability of  $z_k(x) = q, q \in \{0, 1\}$ , is

$$P(z_k(x) = q|y) = \sum_{x' \in \mathcal{X}_k^z} p(y|x') \prod_{\substack{j=0 \\ j \neq k}}^{m-1} \frac{e^{z_j(x')L_a(z_j)}}{1 + e^{L_a(z_j)}} \quad (3.2)$$

where  $\mathcal{X}_k^q$  is the subset of  $\mathcal{X}$  containing those signals whose  $k^{th}$  bit positions are labeled with  $q$ . For the given channel with additive Gaussian noise, the conditional probability of  $y$  given  $x$  is

$$p(y|x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(y-ax)^2}{2\sigma^2}} \quad (3.3)$$

where  $\sigma^2 = N_0/2$  is the noise variance,  $N_0$  is the one-sided power spectral density of the white Gaussian noise, and  $a$  is the fading coefficient (equal to 1 in an AWGN channel).

The demodulator output may be expressed in the form of LLRs by combining (3.2) with (3.3) after several simplifications,

$$\begin{aligned}
L_e(z_k) &= \ln \frac{\sum_{x' \in \mathcal{X}_k^1} p(y|x') \prod_{\substack{j=0 \\ j \neq k}}^{m-1} \exp(z_j(x') L_a(z_j))}{\sum_{x' \in \mathcal{X}_k^0} p(y|x') \prod_{\substack{j=0 \\ j \neq k}}^{m-1} \exp(z_j(x') L_a(z_j))} \\
&= \ln \frac{\sum_{x' \in \mathcal{X}_k^1} \exp \left( -\frac{(y-ax')^2}{N_0} + \sum_{\substack{j=0 \\ j \neq k}}^{m-1} z_j(x') L_a(z_j) \right)}{\sum_{x' \in \mathcal{X}_k^0} \exp \left( -\frac{(y-ax')^2}{N_0} + \sum_{\substack{j=0 \\ j \neq k}}^{m-1} z_j(x') L_a(z_j) \right)}. \tag{3.4}
\end{aligned}$$

The above computation could be facilitated by replacing  $\ln(\sum_i e^{x_i})$  with the max-star operator [52].

### 3.2.2 The Shaping Decoder

The first LLR stream  $L_e(\mathbf{s}_1)$  is de-interleaved and fed in to the shaping decoder as  $L_a(\mathbf{c})$ . The shaping decoder outputs the extrinsic LLRs  $L_e(\mathbf{d})$  and  $L_e(\mathbf{c})$  based on the input from the demodulator and the extrinsic information fed back from the channel decoder. The implementation of the shaping decoder is similar to that of the demodulator, but the summations are now over subsets of the shaping code rather than subsets of the signal constellation. The exponents of the transition probabilities are found by taking the inner product of the  $n_s$  bit LLRs with all the candidate codewords. Taking into account these differences, the output

of the MAP decoder for the shaping code is

$$L_e(d_j) = \ln \frac{\sum_{\mathbf{d}' \in \mathcal{D}_j^1} \exp \left( \sum_{n=1}^{n_s} c_n(\mathbf{d}') L_a(c_n) + \sum_{\substack{\ell=1 \\ \ell \neq j}}^{k_s} d'_\ell L_a(d_\ell) \right)}{\sum_{\mathbf{d}' \in \mathcal{D}_j^0} \exp \left( \sum_{n=1}^{n_s} c_n(\mathbf{d}') L_a(c_n) + \sum_{\substack{\ell=1 \\ \ell \neq j}}^{k_s} d'_\ell L_a(d_\ell) \right)} \quad (3.5)$$

where  $\mathcal{D}_j^t$  denotes the set of messages  $\mathbf{d}$  whose  $j^{\text{th}}$  bit position is labeled with  $t$ ,  $t \in \{0, 1\}$ , and  $c_n(\mathbf{d})$  is the  $n^{\text{th}}$  bit in the codeword associated with message  $\mathbf{d}$ .

The extrinsic information  $L_e(c)$  produced by the shaping decoder can be implemented in a similar manner,

$$L_e(c_j) = \ln \frac{\sum_{\mathbf{c}' \in \mathcal{C}_j^1} \exp \left( \sum_{n=1}^{k_s} d_n(\mathbf{c}') L_a(d_n) + \sum_{\substack{\ell=1 \\ \ell \neq j}}^{n_s} c'_\ell L_a(c_\ell) \right)}{\sum_{\mathbf{c}' \in \mathcal{C}_j^0} \exp \left( \sum_{n=1}^{k_s} d_n(\mathbf{c}') L_a(d_n) + \sum_{\substack{\ell=1 \\ \ell \neq j}}^{n_s} c'_\ell L_a(c_\ell) \right)} \quad (3.6)$$

where  $\mathcal{C}_j^t$  denotes the shaping codewords  $\mathbf{c}$  whose  $j^{\text{th}}$  bit position is labeled with  $t$ ,  $t \in \{0, 1\}$ , and  $d_n(\mathbf{c})$  is the  $n^{\text{th}}$  bit in the message associated with codeword  $\mathbf{d}$ . Note that there are more zeros than ones in the shaping codewords, i.e.,  $|\mathcal{C}_j^0| > |\mathcal{C}_j^1|$ , and therefore the expression in the denominator of (3.6) has more terms than that in the numerator. This is in contrast with (3.5), which has the same number of terms in the numerator and denominator since the message bits are equally likely to be 0 or 1.

### 3.2.3 Receiver with Turbo Decoder

If turbo code is used as the channel code, the soft output  $L_a(\mathbf{u})$  of bit combination P/S will be passed into the turbo decoder. The turbo decoder consists of two SISO convolutional decoders, which are shown in Fig. 3.3 and they are working in the iterative manner [10]. The first SISO decoder (D1) receives the systematic data information  $L_a(\mathbf{u}^{(k)})$ , parity data information of  $\mathbf{u}^{(1)}$   $L_a(\mathbf{u}^{(1)})$  as well as the extrinsic information  $L_a(\mathbf{u}^{(2)})$  from the second



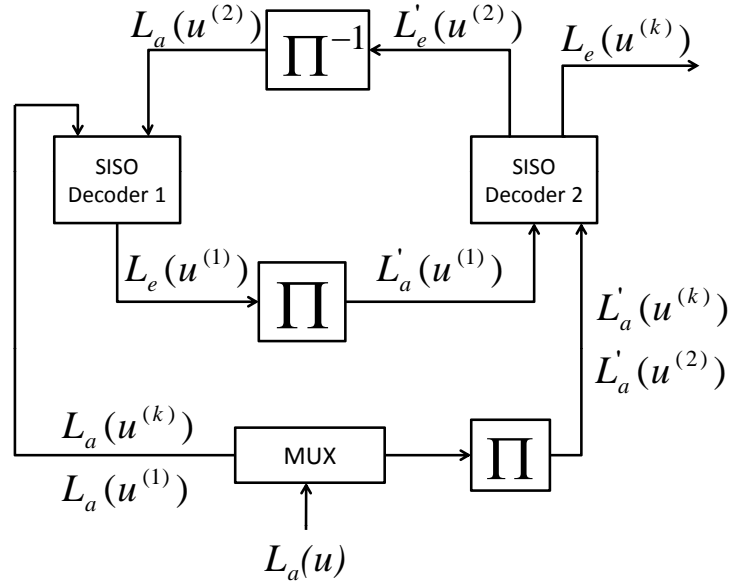


Figure 3.3: Turbo decoder

SISO decoder (D2). Initially,  $L_a(\mathbf{u}^{(2)})$  is set to be all zeros and it is updated during the decoding. D1 performs the BCJR algorithm and produces:

$$L_e(u_i^{(k)}) = \log \frac{p(u_i^{(k)} = 1 | L_a(\mathbf{u}^{(1)}), L_a(\mathbf{u}^{(k)}), L_a(\mathbf{u}^{(2)}))}{p(u_i^{(k)} = 0 | L_a(\mathbf{u}^{(1)}), L_a(\mathbf{u}^{(k)}), L_a(\mathbf{u}^{(2)}))} \quad (3.7)$$

The extrinsic information is calculated by

$$L_e(\mathbf{u}^{(1)}) = L_e(\mathbf{u}^{(k)}) - L_a(\mathbf{u}^{(1)}) - L_a(\mathbf{u}^{(k)}) - L_a(\mathbf{u}^{(2)})$$

$L_e(\mathbf{u}^{(1)})$  is then interleaved to be  $L'_a(\mathbf{u}^{(1)})$ , and also the demultiplexer chooses the proper message  $L'_a(\mathbf{u}^{(k)})$ ,  $L'_a(\mathbf{u}^{(2)})$  to forward into the second SISO decoder (D2) to produce  $L'_e(\mathbf{u}^{(2)})$ . Then the hard decision is produced to test if the bits estimation is correct or the stopping criteria is satisfied. Otherwise the extrinsic information of D2 is calculated and passed into D1 to continue the iterative process.

The turbo code with constraint length 4 defined in UMTS standard is adopted as one of the channel coding schemes. The UMTS standard turbo code has well-designed interleaver and wide range of coding rate and length options [50]. The information data length from 40 to 5114 is supported according to the standard. The base rate of the UMTS code is 1/3, but the standard has a rate matching procedure which support any rate above 1/3.

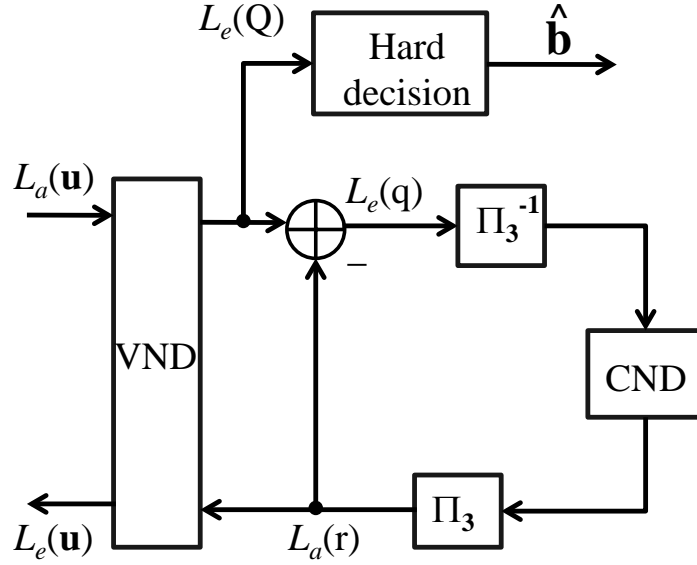


Figure 3.4: LDPC decoder

### 3.2.4 Receiver with LDPC Decoder

Rather than illustrating the decoder structure using a Tanner or factor graph, Fig. 3.4 interprets the LDPC code as the serial concatenation of an inner repetition code and outer single parity check code, i.e. as an extended irregular repeat-accumulate (eIRA) code [28]. Such a representation structure makes the LDPC decoder similar to turbo decoder and helps to illuminate the interaction between the LDPC and shaping decoders. The LDPC decoder is comprised of two processing blocks: a variable node decoder (VND) and a check node decoder (CND) [53]. The VND produces a posteriori messages  $L_e(Q_i)$  of bits  $u_i$ , which will form the input to the CND after interleaving and subtraction by the CND output from previous iteration. The interleaver, denoted by  $\Pi_3$  in the diagram, corresponds to the edges in the code's Tanner graph that connect the variable and check nodes. After interleaving, the output of the CND is used as the a priori input to the VND. Iterative decoding is performed by passing interleaved messages between the VND and the CND until the correct codeword is found or the maximum number of iterations is reached. Each check node in the CND represents one parity-check equation; all parity check equations should be satisfied when the codeword is correctly decoded.

A variable node  $i$  with degree  $d_{u_i}$ , which means it connects to  $d_{u_i}$  check nodes, processes

information coming in from all connected check nodes and from the output of the demapper. The VND for variable node  $i$  simply performs the summation of total  $d_{u_i} + 1$  input messages:

$$L_e(Q_i) = L_a(u_i) + \sum L_a(r_{ji}) \quad (3.8)$$

where  $L_a(r_{ji})$  is the *a priori* LLR coming in variable node  $i$  from the  $j^{th}$  check node, the summation is over the incoming edges. Information  $L_a(u_i)$  is the LLR of code bit  $u_i$  at the input of the LDPC decoder, which normally be the output of the demodulator.  $L_e(Q_i)$  is the LLR that represents the a posteriori probability of the  $i^{th}$  bit  $u_i$  of the LDPC codeword. During each iteration, a hard decision is made on  $L_e(Q)$  to produce an estimate of the codeword and the corresponding message. The decoding process can halt once a correct codeword has been found or the maximum preset iteration number is reached.

The input from variable node  $i$  to the  $j^{th}$  check node is denoted by  $L_e(q_{ij})$ , which could be found by subtracting the message coming from check node  $j$  itself from  $L_e(Q_i)$ ,

$$\begin{aligned} L_e(q_{ij}) &= L_e(Q_i) - L_a(r_{ji}) \\ &= L_a(u_i) + \sum_{j' \neq j} L_a(r_{j'i}), \end{aligned} \quad (3.9)$$

where the summation is over all edges connected to variable node  $i$  except for the one connecting to the  $j^{th}$  check node, i.e. all edges but the one from  $j$   $j' \neq j$ . The subtraction operation in the first line of (3.9) is shown in Fig. 3.2. The CND essentially decodes the single parity-check code associated with each row of the parity-check equation. While there are several ways to formulate the CND decoder, we use a formulation described in [54] which will not be discussed here.

### 3.3 Numerical Results

In this section, we compare the bit error performance of bit-interleaved coded modulation both with and without constellation shaping strategy. The 32-APSK and 16-PAM are considered as the modulation schemes. In the following numerical simulations for shaping systems, *one shaping code* ( $g = 1$ ) for 32-APSK and 16-PAM modulation is focused because of the relatively high shaping gain indicated by the analysis results of chapter 2. Also

the increased system complexity is reasonably small with one shaping code. The following simulation results are grouped according to types of channel coding, i.e., turbo and LDPC code. The system spectral efficiency or system rate  $R$  is chosen by maximizing the shaping gain based on the analysis of section 2.3 and Fig. 2.11. However, since DVB-S2 standard LDPC code is adopted and the limited LDPC code rates defined in the standard may not satisfy this requirement, there may be several parameters adjustments for the LDPC-coded systems. For instance, the information rate is around 3.88 when non-uniform 32-APSK achieves the maximum shaping gain over the AWGN channel, but there is no LDPC code defined in DVB-S2 standard could yield the same spectral efficiency. The parameters chosen for simulation are based on the capacity analysis results but with some modifications for easier system implementation.

### 3.3.1 Results When using Turbo Codes

The used turbo codes are specified in UMTS standard [50]. The UMTS turbo encoder is composed of two constraint length 4 recursive systematic convolutional (RSC) encoders concatenated in parallel [55]. The forward generator is 15 and the feedback generator is 13, both of the numbers are in octal. The number of data bits at the input of the turbo encoder  $k_c$  ranges from 40 to 5114. The length of data bits is set to be 5000, i.e.,  $k_c = 5000$ , for both uniform and shaping system. Without considering the tail bits, the original code rate is  $1/3$ , but the puncturing pattern in UMTS could increase the code rate by removing some of the parity bits. This rate matching operation provides more flexibility to choose the optimal parameters closed to those derived in chapter 2.

The parameters used in the turbo-coded systems are shown in table 3.1, which contains both 16-PAM and 32-APSK modulation schemes with or without shaping. The table shows overall system rate  $R$ , turbo code rate  $R_c$ , shaping code rate  $R_s$ , and value of  $\mathcal{E}_b/N_o$  for which the bit error rate curve reaches  $10^{-5}$ . The situation of  $R_s = 1$  corresponds to uniform system without constellation shaping.

The system receiver contains two receiver implementations. BICM-ID is referred when extrinsic information is fed back from the decoder to the demodulator, BICM is referred

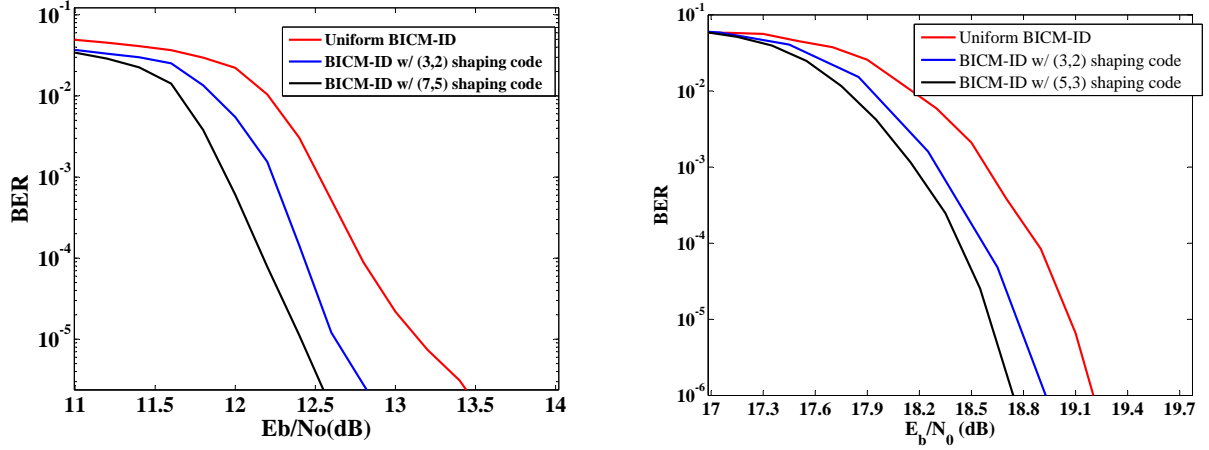
Table 3.1: Parameters used for the systems with turbo code

Mod	channel	$R$	$R_c$	$R_s$	$\mathcal{E}_b/N_0$ in dB
16-PAM	AWGN	2.99	5000/6672	1	13.277
		3.00	5000/6112	2/3	12.695
		3.01	5000/6190	5/7	12.488
	Fading	2.50	5000/8000	1	19.079
		2.50	5000/7337	2/3	18.782
		2.50	5000/7200	3/5	18.604
32-APSK $\{2.64, 4.64\}$	AWGN	3.85	5000/6471	1	6.092
		3.86	5000/6190	7/9	5.834
	Fading	3.00	5000/8320	1	6.261
		2.99	5000/8016	4/5	6.098

when no information is fed back. In BICM-ID receiver for shaping system, information is also exchanged between the turbo and shaping decoders. Two kinds of implementation are used for 32-APSK but only BICM-ID is adopted for 16-PAM since two different shaping codes are considered for 16-PAM modulation. In both receiver structure, we consider 12 decoder iterations, where a full iteration includes an iteration of the shaping decoder, an iteration of the turbo decoder (one iteration for each of the constituent decoders), and, for the case of BICM-ID, one iteration of symbol demapping.

### 16-PAM results

Bit error rate results for 16-PAM over AWGN channel are shown in Fig. 3.5(a), both uniform and shaping systems use BICM-ID model. The system rate  $R = 3$  bits/symbol is considered according to the optimization results. To decide the appropriate combination of turbo code rate  $R_c$  and shaping code rate  $R_s$ , two shaping codes are chosen to approximate the optimal parameter  $p_0 = 0.7287$  in Table 2.1. When the  $R_s = 2/3$  and  $(n_s = 3, k_s = 2)$  shaping code is used, the turbo code rate is set to be  $R_c = 5000/6112$  to ensure the over rate  $R$  is the same as the uniform system. The other combination is using  $(n_s = 7, k_s = 5)$  shaping code and rate  $R_c = 5000/6190$  turbo code. Both the shaping systems perform better than the uniform system with the same system rate. The performance gain, or named shaping



(a) BER of 16-PAM in AWGN at rate  $R = 3.85$  bits/symbol

(b) BER of 16-PAM in ergodic Rayleigh fading at rate  $R = 2.5$  bits/symbol

Figure 3.5: Bit-error rate of 16-PAM in AWGN and fading channel

gain can be found directly from figure or from the last column of Table 3.1, which is 0.58 dB and 0.78 dB when the shaping code is (3, 2) and (7, 5), respectively. The system with (7, 5) shaping code performs better because its  $p_0 = 0.7411$  is closer to the optimal value compared with the system using (3, 2) shaping code with  $p_0 = 0.75$ .

When fading channel is considered, we consider the simple case of Rayleigh fading and the receiver knows the channel information. Fig. 3.5(b) shows the BER curves for uniform and shaping systems at rate  $R = 2.5$ . The chosen  $R$  is based on Table 2.2. Both combinations of shaping code and turbo code mentioned earlier are tested in systems with constellation shaping. The shaping systems perform better than the uniform system, and the system with shaping code (5, 3) have better performance than the other shaping system because of the more optimum  $p_0$  value. The shaping gain achieved in fading channel with  $R = 2.5$  BICM-ID system is up to 0.47 dB.

### 32-APSK results

When 32-APSK modulation is used, we let  $R = 3.85$  bits/symbol for AWGN channel, while for the ergodic Rayleigh fading channel, we consider  $R = 3$  bits/symbol. For each  $R$ , we consider a system with a shaped constellation and a system with a uniform constellation.

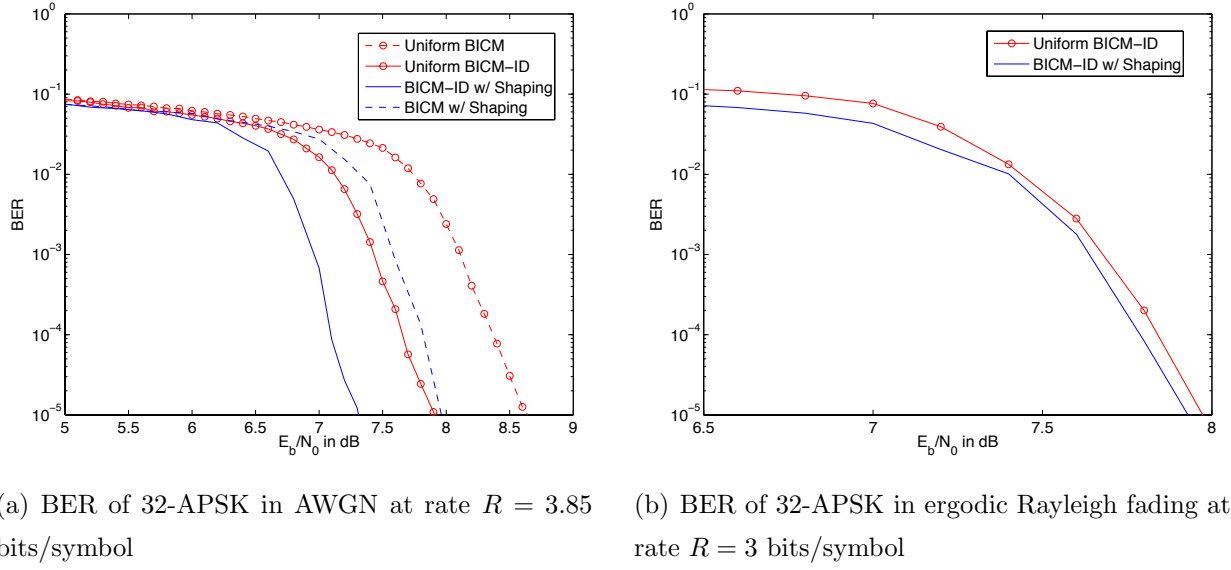


Figure 3.6: Bit-error rate of 32-APSK in AWGN and fading channel

The detail system parameters are listed in table 3.1. These rates were selected because they are close to the rates that achieve the highest shaping gain.

Bit error rate results with and without shaping for the AWGN channel are shown in Fig. 3.6(a). The BICM-ID system model provides a gain of about 0.7 dB over BICM. When using BICM or BICM-ID, both the shaping gain is about 0.6 dB. This shaping gain is actually higher than the 0.265 dB gain predicted by the information-theoretical analysis (see Table 2.1). This additional gain is likely caused by the use of the relatively long shaping code (9, 7) and the inclusion of its decoder in the overall decoding loop. Essentially, the shaping code adds some additional coding gain over the gain of the turbo code.

Results for fading are shown in Fig. 3.6(b). Because of its promising performance, only BICM-ID is considered. The shaping gain in fading is around 0.1 dB, which is smaller than that from the AWGN case but close to the value predicted by the information-theoretic analysis. The reason for the small shaping gain in fading is a subject of ongoing investigation, but it may be due to the fact that the optimal shaping parameters are very sensitive to the SNR, and in a fading channel, the SNR is constantly fluctuating. It is difficult to engineer a system with constellations shaping that is universally good across a broad range of SNRs.

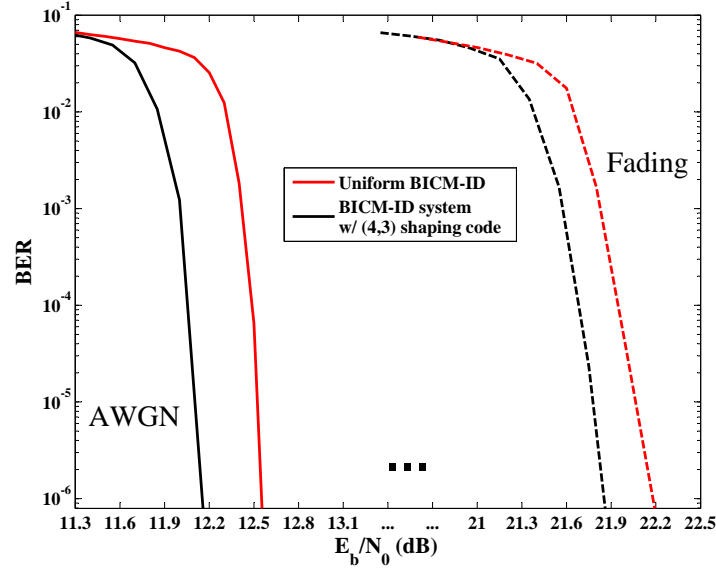


Figure 3.7: Bit-error rate of LDPC-coded 16-PAM at rate  $R = 3$  bits/symbol both with and without shaping. Curves are shown for AWGN and Rayleigh Fading channel.

### 3.3.2 Results when using LDPC Codes

In this part, we compare the bit-error performance of bit-interleaved LDPC coded modulation in uniform and shaping system. The LDPC codes are chosen from the DVB-S2 standard, with possible code rate  $1/4$ ,  $1/3$ ,  $2/5$ ,  $1/2$ ,  $3/5$ ,  $2/3$ ,  $3/4$ ,  $4/5$ ,  $5/6$ ,  $8/9$ , and  $9/10$ . The codeword length is  $n_c = 64,800$ . The overall system rate is selected to be 3 bits/symbol ( $R = 3$ ) for systems with 16-PAM and 32-APSK, which is easy to identify the compatible shaping code rates  $R_s$  with the limited LDPC rates  $R_c$  defined in the standard. The shaping systems use one shaping bit ( $g = 1$ ) which we previously showed to strike a balance between performance and system complexity [44]. The maximum iteration of the receiver is set to be 100, but the decoding operation could be stop earlier if the correct information bits are recovered.

For the rate 3 system ( $R = 3$ ) with 16-PAM modulation, the code rate  $R_c = 3/4$  LDPC code is used in the uniform system; the rate  $R_c = 4/5$  LDPC code combining with shaping code  $R_s = 3/4$  is chosen to ensure the shaping system has the identical system rate as the uniform system. The BER curves are presented in Fig. 3.7, which include systems performance for both AWGN and fading channel. Noticed the SNR values ranging from 13 to 21 on x-axis are omitted because of the space limitation. The shaping system perform



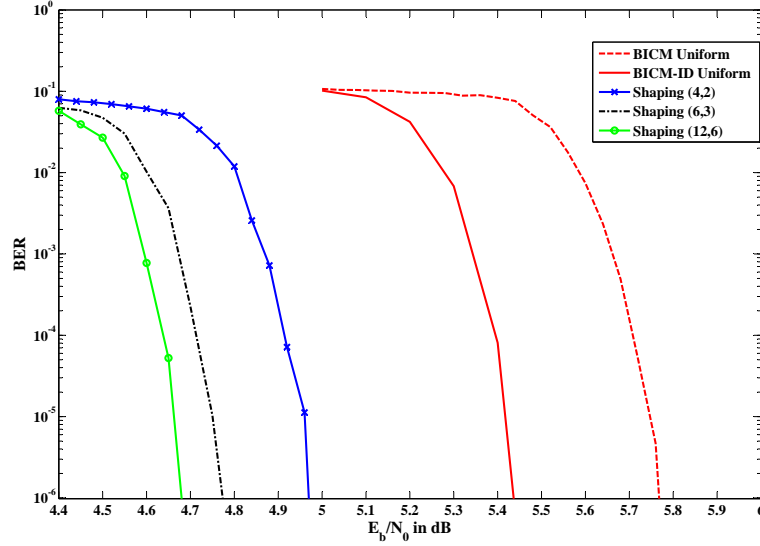


Figure 3.8: Bit-error rate of LDPC-coded 32-APSK in AWGN at rate  $R = 3$  bits/symbol both with and without shaping. Curves are shown for the unshaped (uniform) system using BICM and BICM-ID. Curves are shown for three shaping codes, and the shaped system uses BICM-ID.

better under different channel conditions, the performance gain under AWGN channel and Fading channel is around 0.40 dB and 0.32 dB, respectively.

The APSK modulation used is specified in the DVB-S2 standard [19]. The amplitudes of the 32-APSK constellation are selected such that the middle ring's radius is 2.64 times the inner ring's radius, and the outer ring's radius is 4.64 times the inner ring's radius. To ensure  $R = 3$  both with and without shaping,  $R_c = 3/5$  is used for the uniform (unshaped) system, and  $R_s = 1/2$  and  $R_c = 2/3$  are used for the shaped system. Three different  $R_s = 1/2$  shaping codes are simulated to test if the system performance changes with the increasing of shaping codeword length.

The bit-error rates for AWGN and ergodic (fully-interleaved) Rayleigh fading channels are shown in Figs. 3.8 and 3.9. By comparing the two curves on the right of Fig. 3.8, it is clear that BICM-ID provides a gain of about 0.3 dB over BICM in an AWGN channel without shaping. Comparing the curve for unshaped BICM-ID (second from right) with the curves for shaping (the three curves on the left), it can be seen that shaping provides gains between 0.45 dB to 0.7 dB, depending on the  $R_s = 1/2$  shaping code used. Combining both gains (shaping and BICM-ID gains), a gain of over 1 dB is observed compared to the

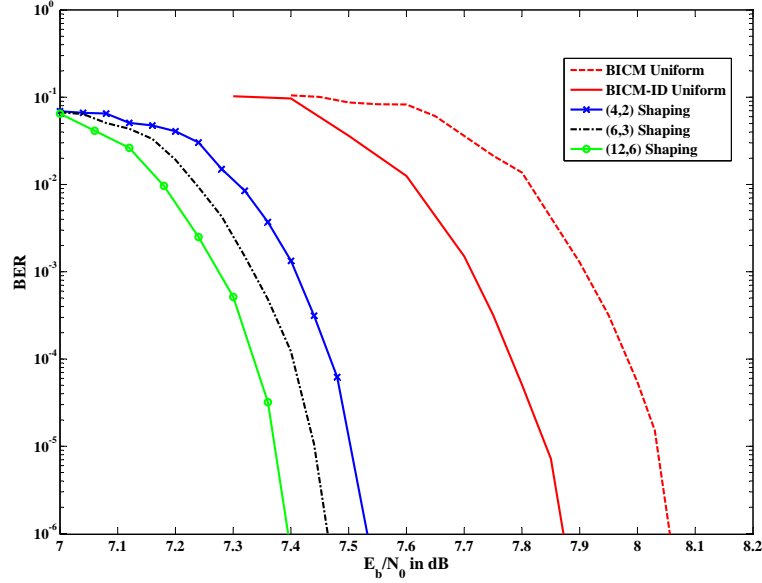
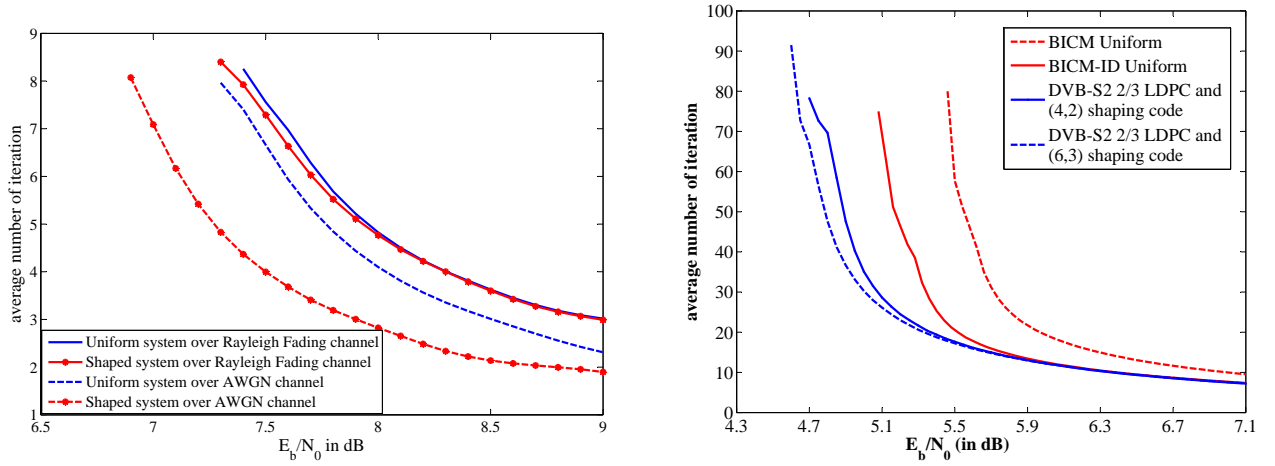


Figure 3.9: Bit-error rate for the same system described in Fig. 3.2 in fully-interleaved Rayleigh fading.

uniform BICM system. Over the Rayleigh fading channel, BICM-ID provides a gain of 0.17 dB over the BICM system, and the additional shaping gain ranges from 0.4 dB to 0.5 dB. Both figures show that longer shaping codewords provide higher shaping gains, at the cost of a more complex decoder. When rate-1/2 shaping is required, the (6, 3) shaping code gives a good tradeoff between performance and complexity. Since such a code only contains  $2^3 = 8$  codewords, its decoding complexity is quite low.

### 3.3.3 Receiver Complexity Consideration

The complexity brought by implementing shaping technique mainly lies on the shaping decoder, in this part we try to numerically compute the complexity difference between the system with uniform constellation and the system with non-uniform constellation. Fig. 3.10(a) shows the average number of iterations required for the shaped and uniform 32-APSK systems to converge (i.e., correct all errors in a frame) in AWGN and fading channel when turbo code is the error control code. This is a useful metric when the receiver operates using an automatic halting mechanism, for instance, performing CRC-based error detection after each iteration. For both channels, the shaped system needs fewer iterations, though the difference is more pronounced in AWGN than in the fading channel. For instance, at



(a) Average number of iterations when turbo code is used

(b) Average number of iterations when LDPC code is used

Figure 3.10: Average number of BICM-ID global iterations required for correcting all code-word errors.

$E_b/N_0 = 8$  dB, which is a typical operating point for the unshaped system, an average of 4.1 iterations are required in AWGN for the unshaped system. At the same operating point, the shaped system only requires about 2.8 iterations to converge. The average iteration number for LDPC coded system are shown in 3.10(b). The same conclusion could be drawn when using LDPC codes. While the per-iteration complexity of the shaped system is higher than that of the uniform system, the need for fewer iterations may result in a lower overall complexity when early halting techniques are used.

The overall and per-iteration complexity of the two systems can be compared by relating the actual execution time of their implementations. The simulation results were produced with software implementations of the systems running on a PC computer with a single-core 3.40 GHz Pentium 4 processor and the Windows XP operating system. For each system, the simulation was executed as fast as possible. The execution time, total number of data bits simulated, and total number of decoder iterations executed were logged. From this information, the throughput of the implementation was quantified in units of bit-iterations per second (bps·iteration). For the simulations over the AWGN channel, the processing rate is 139 kbps·iteration for the shaped system and 170 kbps·iteration for the uniform system. When running over Rayleigh Fading channel, the throughput is 148 kbps·iteration for the

shaped system and 155 kbps-iteration for the uniform system. The processing rate of shaped system over fading channel is greater than that over AWGN channel because of the shorter shaping codeword (the complexity of the shaping decoder is exponential in the length of the shaping encoder's input  $k_c$ ). While shaping reduces the per-iteration throughput, the overall throughput may be reduced due to the need for fewer iterations. For instance, using the previously mentioned turbo code example operating point of  $E_b/N_0 = 8$  dB over the AWGN channel, the uniform system has a throughput of  $170/4.1 = 41.5$  kbps while the shaped system's throughput is  $139/2.8 = 49.6$  kbps.

### 3.4 Conclusion

By incorporating constellation shaping along with iterative demodulation and decoding, an additional coding gain can be attained by using both turbo and LDPC code. This result derived from mutual information analysis is further validated by several simulations with two different modulations and two channel models. One dimension modulation 16-PAM and two dimension 32-APSK are chosen to be the modulation schemes; channel models includes the AWGN and ergodic Rayleigh fading channel. Various system transmission rates are chosen based on the parameters derived from previous chapter. The rate of 3 bits per symbol is considered at LDPC coded system because of the code rates constraints defined in DVB-S2 standard. Performance Gains can be attributed to the feedback structure of the receiver, the constellation shaping strategy, and also the choice of shaping code. Due to the inclusion of a nonlinear shaping code and its corresponding decoder, the proposed system has a higher per-iteration complexity than a standard uniform system. However, by carefully choosing parameters, a shaping code that provides a good tradeoff between performance and complexity can be identified.

## Chapter 4

# EXIT Chart and LDPC Code Optimization

Extrinsic information transfer (EXIT) charts visualize the iterative decoding process and predict the convergence *threshold* value by analyzing the input/output relations of individual demodulator or decoder [53, 56]. The relation is the mutual information between the input and output of the demodulator/decoder. In this chapter, a simple introduction of EXIT charts and the generating methods are presented. Then EXIT charts are used as the guide to analysis both the uniform and shaped BICM-ID system. The optimized LDPC code degree distributions are found based on the EXIT charts analysis and the newly developed codes are adopted in the system mentioned in the previous chapter. Numerical simulation results are presented at the end of this chapter.

### 4.1 EXIT Charts Based Analysis

#### 4.1.1 Exit Charts for Uniform System

An EXIT chart typically consists of two curves. One is the demodulator mutual information transfer characteristic and the other one is the decoder mutual information transfer characteristic. Each transfer characteristic is the plot of mutual information at the output of the demodulator/decoder as a function of the mutual information of the a priori information at the input side.

Two assumptions obtained by simulation are used to generate EXIT curves. For large interleavers the a priori messages remain fairly uncorrelated from the input information of demodulator/decoder after many iterations. The probability density function of the output value of demodulator/decoder approaches Gaussian-like distribution with the increasing of iteration number [56]. According to these two observations, the a priori input  $A$  to the demodulator/decoder can be modeled as Gaussian random variable with zero mean and variance  $\sigma_A^2$  in conjunction with the known transmitted bits  $\mathbf{X}$  [57, 56].

$$A = \mu_A \cdot \mathbf{X} + n_A \quad (4.1)$$

Since  $A$  is assumed to be Gaussian distributed and the conditional probability density function of the a priori input  $A$  can be written as:

$$f_A(\xi|\mathbf{X} = x) = \frac{\exp\{-(\xi - \mu_A \cdot x)^2/2\sigma_A^2\}}{\sqrt{2\pi}\sigma_A} \quad (4.2)$$

Note that the true LLR with symmetric distributions satisfies the *consistency condition* [58]:

$$f_A(\xi|x) = f_A(-\xi|x) \cdot e^{x \cdot \xi}, \quad (4.3)$$

the mutual information between transmitted bits  $\mathbf{X}$  and the  $L$ -value  $A$  is derived based on the expressions above:

$$I_A(\sigma_A) = 1 - \int_{-\infty}^{\infty} \frac{\exp(-\frac{(\xi - (\sigma_A^2/2))^2}{2\sigma_A^2})}{\sqrt{2\pi}\sigma_A} \log_2(1 + \exp(-\xi)) d\xi. \quad (4.4)$$

Equation (4.4) is also defined as ‘J’ function

$$J(\sigma) = I_A(\sigma_A = \sigma). \quad (4.5)$$

This function is monotonically increasing and thus reversible:

$$\sigma_A = J^{-1}(I_A) \quad (4.6)$$

The mutual information  $I_E$  is used to quantify the mutual information of demodulator/decoder extrinsic output, which is the function of  $I_A$  and  $\mathcal{E}_s/N_0$  and can be defined as  $I_E = T(I_A, \mathcal{E}_s/N_0)$ . The Monte Carlo simulation is the most convenient way to determine the value

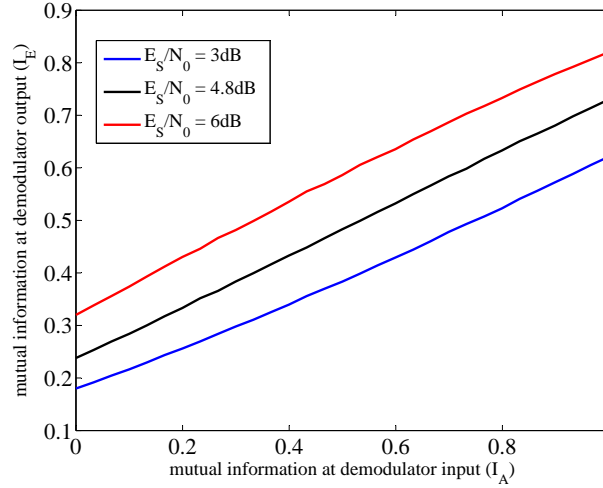


Figure 4.1: Demodulator mutual information transfer characteristics for 16-QAM with 'SP' labeling

of  $I_E$  for the desired  $(I_A, \mathcal{E}_s/N_0)$  combination. The independent Gaussian random variable of (4.1) is applied as a priori input to the demodulator or decoder of interest,  $\sigma_A$  is chosen appropriately by the certain value of  $I_A$  according to the reverse function  $J^{-1}$ .

To plot the curve of demodulator transfer characteristic, a long sequence of bits  $\mathbf{X}$  are generated and mapped into constellation symbols, which are transmitted through the desired channel with certain signal to noise ratio  $\mathcal{E}_s/N_0$ . The Gaussian distributed a priori LLRs  $L_A$  with mutual information  $I_A$  are generated such that the variance  $\sigma^2$  satisfies the equation (4.6). The mutual information of the demodulator output is

$$I_E = 1 - \frac{1}{\log(2)\log_2(M)} \sum_{i=1}^{\log_2 M} E[\max^*(0, L_{E_i}(-1)^{x_i})] \quad (4.7)$$

where  $L_{E_i}$  is the extrinsic output information of the demodulator in log-domain and  $I_E$  is found by using Monte Carlo simulation by averaging the results of large number of trials. Fig. 4.1 shows demodulator transfer characteristics for 16-QAM under several  $E_s/N_0$  values. The a priori inputs  $I_A$  are on the abscissa, the extrinsic outputs  $I_E$  are on the ordinate. The  $\mathcal{E}_s/N_0$ -value serves as a parameter to the curves. The value of  $I_E$  when  $I_A = 0$  is the BICM capacity. At higher SNR point, the demodulator transfer curve lies above the one generated at lower SNR value since the demodulator output contains more mutual information. The demodulator transfer characteristics are also different when the constellation mapping rules

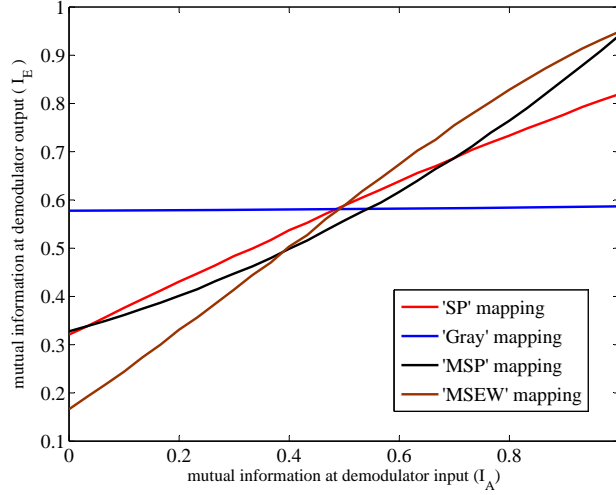


Figure 4.2: Demodulator mutual information transfer characteristics for different 16-QAM mappings at  $\mathcal{E}_s/N_0 = 4.5\text{dB}$

are not identical. The curves in Fig. 4.2 are 16-QAM demodulator characteristics generated based on four different labeling rules at the same SNR value  $\mathcal{E}_s/N_0 = 6.5\text{dB}$ . The Gray mapping does not provide any gain when the a priori mutual information is increasing. By using other mappings the output of demodulator leads to a higher extrinsic mutual information, however, the increasing trends are different. The detail discussion about the constellation labeling rule will be presented in the next chapter.

For the decoder transfer characteristics curve, the Gaussian distributed a priori input  $\lambda \sim \mathcal{N}((2x-1)\sigma^2/2, \sigma^2)$  is generated as the input to the decoder. The mutual information of the decoder output is a function of the a priori input message, which can be derived by the same equation (4.7). The decoder transfer characteristics is not related to the channel signal to noise ratio since the a priori information is assumed to be gaussian distributed with known variance. Fig. 4.3 shows the decoder transfer characteristic for rate 1/2 convolutional codes with different generators and constraint length  $K$ . The code polynomials are expressed in octal number. The transfer characteristics  $I_E$  are monotonically increasing with respect to the value of  $I_A$  according to Fig. 4.1 and 4.3. The larger value of  $I_A$  means more bits are known at the demodulator/decoder with high confidence, which will provide extrinsic output with increasing mutual information.

Based on the structure of the iterative receiver in Fig. 3.2, messages are exchanged



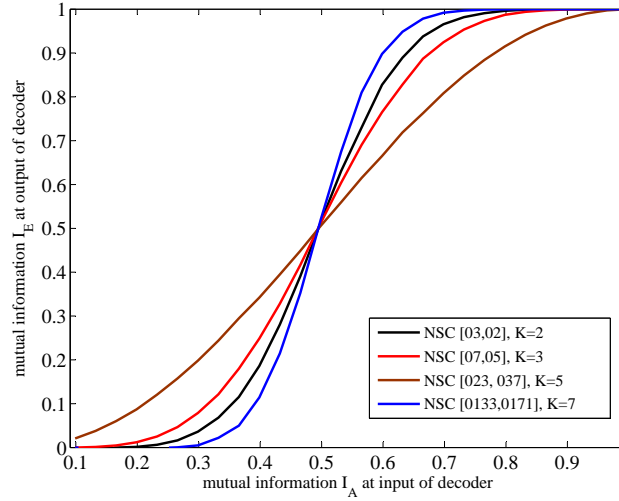


Figure 4.3: Decoder mutual information transfer characteristics for rate 1/2 convolutional code with different code memory

between the a priori inputs and the extrinsic outputs. Let  $i$  be the iteration index and iterative decoding starts at the origin ( $i = 0$ ) with zero a priori information  $I_{A_{dem,0}} = 0$ . The output of the demodulator becomes the input to the decoder after interleaving, and also the output of the decoder is deinterleaved to be the input of the demodulator. Notice that the interleaver and deinterleaver have no impact on the mutual information. When it comes to the  $i$ -th iteration, the extrinsic output of the demodulator  $I_{E_{dem,i}}$  is forwarded to the decoder as the a priori input  $I_{A_{dec,i}} = I_{E_{dem,i}}$ . The output of the decode with mutual information  $I_{E_{dec,i}} = f(I_{A_{dec,i}})$  becomes the a priori knowledge  $I_{A_{dem,i+1}} = I_{E_{dec,i}}$  of the demodulator at next iteration. The message exchange process between the demodulator and decoder can be visualized by plotting the two transfer characteristics into a single diagram. The axes need to be swapped for the decoder transfer characteristics. The iteration process stops if there is an intersection of the two characteristics curves.

The EXIT curves for 16-QAM with convolutional code at  $\mathcal{E}_s/N_0 = 6$  dB are shown in Fig. 4.4. The trajectory of iterative decoding has managed to pass through the space between the two curves, which indicates the high possibility of the successful decoding process with the increasing of the iterations at this SNR value. The bottleneck region of the EXIT chart becomes smaller along with the decreasing of the  $\mathcal{E}_s/N_0$ -value. The minimum  $\mathcal{E}_s/N_0$  value

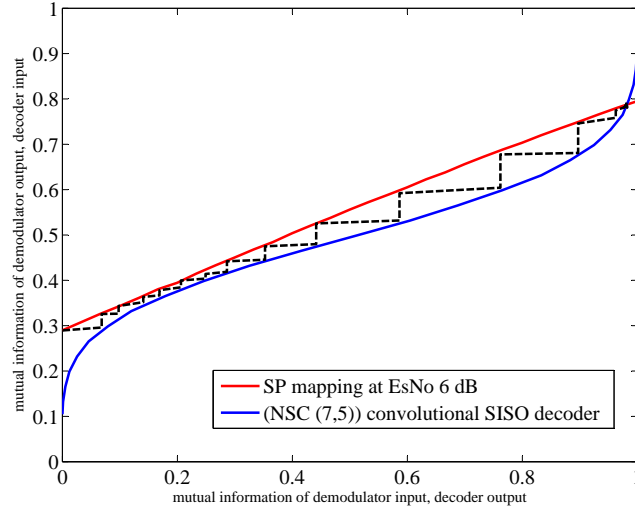


Figure 4.4: EXIT chart for 16-QAM modulation with 'SP' mapping at  $\mathcal{E}_s/N_0 = 6\text{dB}$ , averaged decoding trajectories are also shown

required to raise the demodulator curve above the decoder curve is the convergence *threshold*, which corresponds to the turbo cliff region of the BER curve. The threshold for the uniform system introduced in chapter 3 with 32-APSK modulation and turbo code is 7.5 dB, while the simulation results in Fig. 3.6(a) shows the SNR is 7.9 dB when the BER curve touches  $10^{-5}$ . The advantage of EXIT chart is that only simulations for individual demodulator or decoder is needed to predict the behavior of the iterative decoder, no intensive error rate simulations of the iterative decoding scheme are required.

### EXIT Charts for Uniform Systems with LDPC Code

The EXIT chart contains two curves for LDPC-coded systems, one curve is the transfer characteristic of the combined demodulator and LDPC variable node decoder (VND), the other one is the characteristic of the check node decoder (CND) [53]. In the non-uniform system with constellation shaping, the “demodulator” is not only responsible for demapping the modulated symbols and also decoding shaping codewords, so the curve that combines the demodulator and VND in uniform system will also include the shaping code decoder in non-uniform system. The CND part is similar for both uniform and shaping system.

The decoding algorithm of variable node part of LDPC code is

$$L_{i,\text{out}} = L_{\text{ch}} + \sum_{j \neq i} L_{j,\text{in}} \quad (4.8)$$

$L_{j,\text{in}}$  is the  $j$ th a priori message received by the variable node,  $L_{i,\text{out}}$  is the extrinsic message comes out of the variable node,  $L_{\text{ch}}$  is the message from the channel. When model the a priori input of the VND as the Gaussian distribution with variance  $(J^{-1}(I_A))^2$ , the EXIT function of a degree- $d_v$  variable node is

$$I_{E,\text{VND}} \left( I_A, d_v, \frac{E_b}{N_0}, R \right) = J \left( \sqrt{(d_v - 1)[J^{-1}(I_A)]^2 + \sigma_{\text{ch}}^2} \right) \quad (4.9)$$

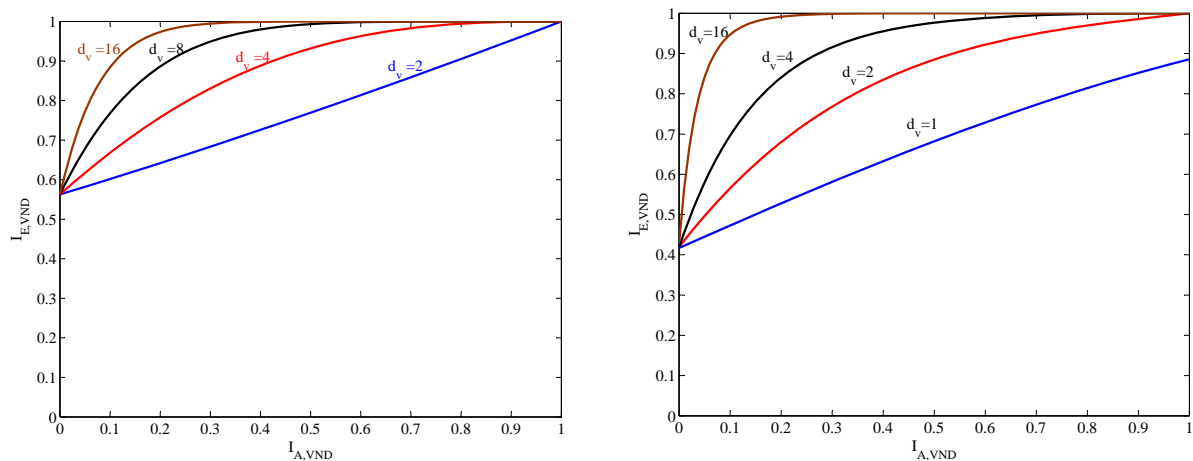
where the function  $J(\cdot)$  and  $J^{-1}(\cdot)$  are given in equation (4.4) and also described in the Appendix of [53]. The variance of the channel message  $L_{\text{ch}}$  is related to system rate  $R$  and signal noise ratio  $\mathcal{E}_b/N_0$ . Assume the situation of AWGN channel and BPSK modulation, the numerical representation of the variance of channel message is  $\sigma_{\text{ch}}^2 = 4/\sigma_n^2 = 8R \cdot (E_b/N_0)$ . Fig. 4.5(a) shows the VND curves of rate 1/2 LDPC codes with different variable node degrees combined with BPSK modulation. While in circumstances other than BPSK modulation the variance  $\sigma_{\text{ch}}^2$  cannot be expressed in closed form and Monte Carlo simulation is used to determine its value.

The methodology of generating the CND EXIT curves are the same for both uniform and shaping system. The expression can be found in [53] and we quote it here:

$$I_{A,\text{CND}}(I_E, d_c) \approx 1 - J \left( \frac{J^{-1}(1 - I_E)}{\sqrt{d_c - 1}} \right) \quad (4.10)$$

Fig. 4.6 plots several check node curves with different degrees, the shape of the curves is similar to the VND curves in 4.5(a) if the axes are swapped.

The detector curve without shaping decoder is generated in the conventional method. A length- $N_c$  sequence of simulated code bits  $\mathbf{z}$  is generated randomly (according to a uniform distribution) and passed through the modulator to produce the modulated symbols  $\mathbf{x}$ . The received symbols  $\mathbf{y}$  is found by adding the modulated symbols  $\mathbf{x}$ , which have average energy  $\mathcal{E}_s$ , to zero-mean circularly symmetric complex Gaussian noise with variance  $N_0/2$  in each complex dimension. The demodulator receives the vector  $\mathbf{y}$  along with a vector of generated *a priori* LLRs  $L_a(\mathbf{z})$ , which represent the interleaved information fed back from the LDPC



(a) Variable Node Decoder Exit curve at  $E_b/N_0 = 1$  dB and code rate  $R = 1/2$

(b) Combined 16-QAM detector and VND EXIT curves at  $E_b/N_0 = 5$  dB and code rate  $R_c = 1/2$

Figure 4.5: VND curves for LDPC codes over AWGN channel

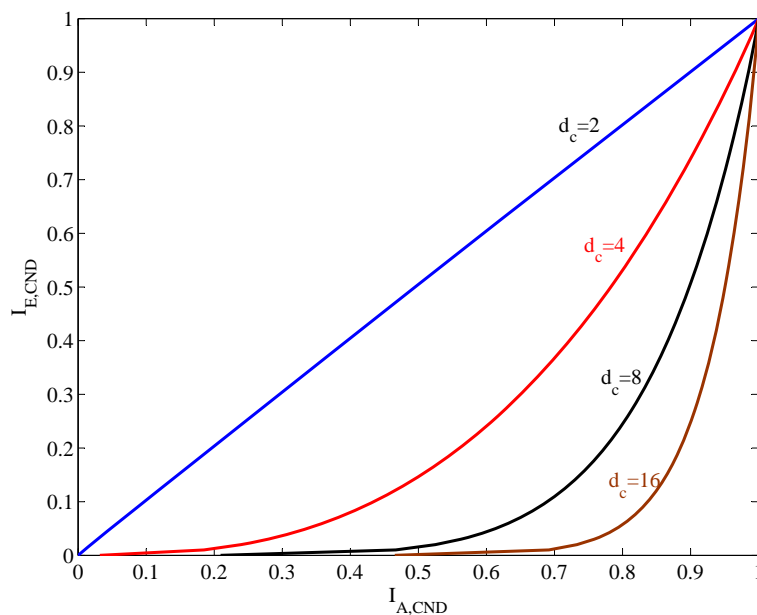


Figure 4.6: CND EXIT curves

decoder. Elements of  $L_a(\mathbf{z})$  is Gaussian distributed with variance  $\sigma_a$ , which is related to  $I_A$  by equation (4.4), and the mean value equal to  $\sigma_a^2/2$  when the corresponding code bit in  $\mathbf{z}$  is 1 and  $-\sigma_a^2/2$  when the corresponding code bit is 0. Using  $\mathbf{y}$  and  $L_a(\mathbf{z})$  as inputs, the extrinsic information  $L_e(\mathbf{z})$  at the output of the demodulator is found by executing the demodulator. For each value of  $\mathcal{E}_s/N_0$  and  $I_A$ , this process is repeated a large number of Monte Carlo trials to get the numerically averaged  $I_E$ .

We denote the EXIT transfer characteristic of the detector by  $I_{E,\text{DET}}$ , which is the function of  $I_{A,\text{DET}}$ ,  $R$  and  $\frac{E_b}{N_0}$ . This curve is then approximated by a third-order (or higher) polynomial  $f(I_{A,\text{DET}})$  after the estimation process, with the goal of combining detector and VND curve by using equation (4.9) with  $\sigma_{\text{ch}} = J^{-1}(I_{E,\text{DET}})$ . The combined EXIT curve is denoted by

$$I_{E,\text{VND}} \left( I_{A,\text{VND}}, d_v, \frac{E_b}{N_0}, R \right) \quad (4.11)$$

The a priori message feedback to the detector with the mutual information  $I_{A,\text{DET}}$  is the sum of the messages passing to the variable node ( $\sum_j L_{j,\text{in}}$ ), the mutual information of which is  $I_{A,\text{VND}}$ .

$$I_{A,\text{DET}}(I_{A,\text{VND}}, d_v) = J \left( \sqrt{d_v} \cdot J^{-1}(I_{A,\text{VND}}) \right) \quad (4.12)$$

The last step is to insert the expressions described above into equation (4.9) and the combined detector/VND EXIT curve is in the form

$$I_{E,\text{VND}}(I_A, d_v, I_{E,\text{DET}}) = J \left( \sqrt{(d_v - 1)[J^{-1}(I_{A,\text{VND}})^2] + [J^{-1}(I_{E,\text{DET}})]^2} \right) \quad (4.13)$$

Fig. 4.5(b) plots several combined 16-QAM detector and VND transfer curves, note that when  $d_v = 1$  the pure demodulator transfer characteristic is recovered for 16-QAM modulation at  $\mathcal{E}_b/N_0 = 5$  dB. The EXIT chart for the DVB-S2 standard LDPC with 32-APSK modulation is shown in Fig. 4.7. The two curves are tangent so the SNR value at which the figure is plotted is also the convergence threshold.

### 4.1.2 Exit Charts for Shaped System

Considering the system with constellation shaping, we use the system model of Fig. 3.2 as the example. The “detector” actually performs two operations: 1) demapping of the

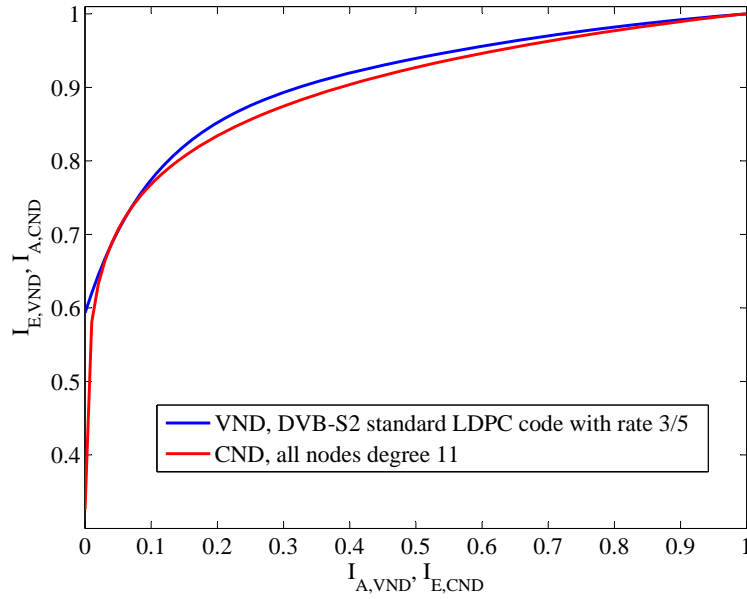


Figure 4.7: Exit chart for the uniform system with DVB-S2 standard LDPC code and 32-APSK modulation at  $\mathcal{E}_b/N_0 = 4.93$  dB

corrupted constellation symbols, and 2) decoding of the shaping code. The combination of shaping code and modulation can be interpreted as a high-dimensional modulation, i.e. an order- $2n_s$  modulation, since there will be  $n_s$  two-dimensional APSK signals selected per shaping codeword. The “decoder” part remains unchanged as that in uniform system for recovering data from turbo or LDPC codeword.

The mutual-information transfer characteristic curve for the combined detector is generated using a Monte Carlo method. A length- $N_c$  sequence of simulated code bits  $\mathbf{v}$  is generated randomly (according to a uniform distribution) and passed through the transmitter shown in Fig 2.5. The shaped codeword  $\mathbf{z}$  are produced and fed into the modulator to be the sequence of modulated symbols  $\mathbf{x}$ . For the AWGN channel, the vector of received symbols  $\mathbf{y}$  is found by adding the modulated symbols  $\mathbf{x}$  to zero-mean Gaussian noise with variance  $N_0/2$  in each complex dimension. The “detector”, which actually contains both the demodulation and the shaping decoder, receives at its input the vector  $\mathbf{y}$  along with a generated vector *a priori* LLRs  $L_a(\mathbf{v})$  as the fed back information from the channel code decoder. Each element of  $L_a(\mathbf{v})$  is Gaussian random variable with variance  $I_{A,DET}$ , which is related to  $\sigma_a$  by equation (4.4). The mean value equals to  $\sigma_a^2/2$  when the corresponding code bit in  $\mathbf{v}$  is 1 and  $-\sigma_a^2/2$

when the corresponding code bit is 0. The extrinsic information  $L_e(\mathbf{z})$  at the output of the demodulator is found by executing an iteration of the shaping decoder and APSK demapper along with the required interleaving and bit separation. For each value of  $\mathcal{E}_s/N_0$  and  $I_a$ , this process is repeated a large number of Monte Carlo trials. For the  $k^{th}$  Monte Carlo trial, The mutual information between  $\mathbf{v}$  and the output of the demodulator is:

$$I_e[k] = 1 - \frac{1}{N_c} \sum_{n=1}^{N_c} \log_2 (1 + \exp^{-v_n L_e(v_n)}) \quad (4.14)$$

where  $v_n$  is the  $n^{th}$  element of  $\mathbf{v}$ . If  $L$  trials are executed, then the Monte Carlo estimate of the average mutual information is:

$$I_e = \frac{1}{L} \sum_{k=1}^L I_e[k]. \quad (4.15)$$

The mutual-information transfer characteristic for the turbo decoder is to generate the mutual information of its output  $L_e(\mathbf{u})$  as the function of the input mutual information related to  $L_a(\mathbf{u})$ . Instead of using generated Gaussian a priori inputs to generate this curve, we use the more accurate method proposed for turbo-coded BICM-ID in [52]. The method involves using the actual output from the demodulator as the input to the turbo decoder, rather than using simulated values. The mutual information of the decoder output  $L_e(\mathbf{u})$  was tracked by simulating the turbo codewords at each decoder iteration. This process is repeated for many long turbo codewords, and the average mutual information of the decoder is plotted for each iteration against the output extrinsic information of the shaping decoder.

Fig. 4.8 shows the EXIT chart example of the shaping system using 32 APSK modulation at system rate 3.85 over  $\mathcal{E}_b/N_0 = 7$  dB AWGN channel. In this figure,  $I_e(\mathbf{u})$  and  $I_a(\mathbf{u})$  are used to denote the mutual information at the input and output of the turbo decoder, respectively. Similarly,  $I_e(\mathbf{v})$  and  $I_a(\mathbf{v})$  are used to denote the mutual information at the input and output of the turbo decoder. Multiple decoder transfer characteristic curves correspond to different turbo iterations, and curves beyond the sixth iteration are not shown since they are not significantly different from that of sixth iteration. The demodulator characteristic curve with shaping is steeper than it is without, which suggests that shaping code may provide an additional coding gain.

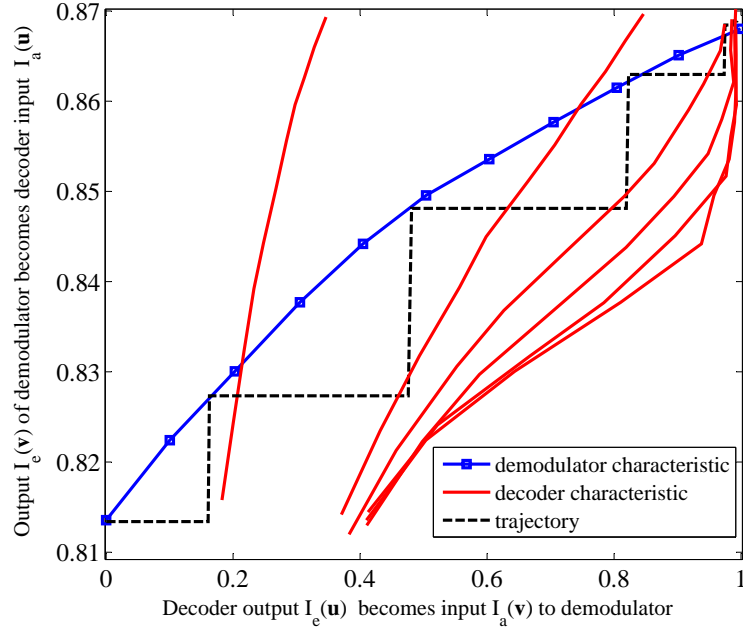


Figure 4.8: EXIT chart for BICM-ID using M=32 APSK and overall system rate  $R=3.85$  over AWGN channel at  $\mathcal{E}_b/N_0 = 7\text{dB}$ . Different decoder characteristic curves correspond various channel decoder iterations. From left to right, the six decoder curves correspond to 1 through 6 iterations.

The trajectories of the iterative decoding are also shown in Fig. 4.8 to illustrate the convergence behavior of the BICM-ID receiver. The multiple decoder curves are used to determine the trajectories as follows. During the first iteration, the mutual information at the output of the demodulator is the value of the characteristic for which  $I_a(\mathbf{v}) = 0$ . The trajectory moves to the right until it encounters the first-iteration decoder characteristic curve, at which point the trajectory moves upward. It proceeds upward until it encounters the demodulator characteristic curve, at which point it moves to the right. Now, because we are dealing with the second iteration, the trajectory moves to the right until it encounters the *second*-iteration decoder curve. The process continues, with the trajectory reaching the next decoder curve after each iteration. The decoding will succeed with high probability if the trajectory proceeds to the right side of EXIT chart, indicating that the value of  $\mathcal{E}_s/N_0$  is greater than the threshold. Otherwise if the trajectory gets stuck at some point  $I < 1$ , the decoding may not succeed and the current value of  $\mathcal{E}_s/N_0$  is smaller than the threshold. The threshold of the shaped system described in subsection 3.3.1 with turbo code is 6.8 dB



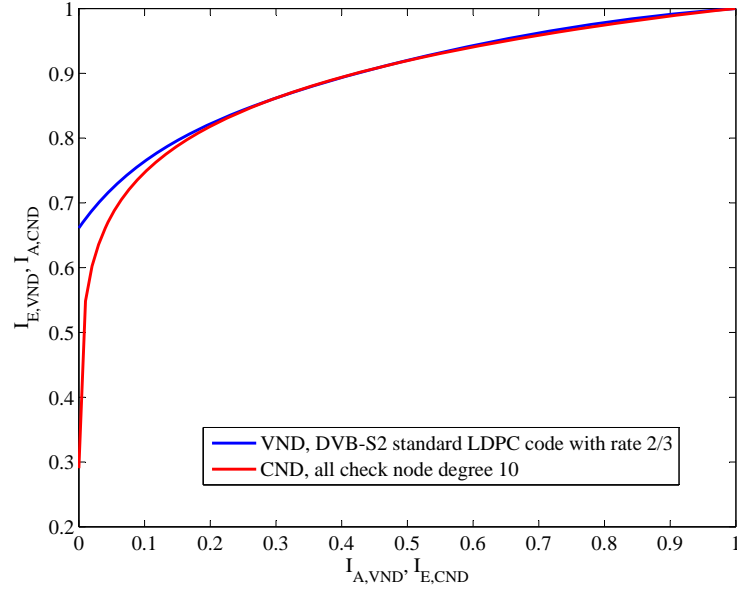


Figure 4.9: EXIT chart for the shaped system described in chapter 3 using M=32 APSK and overall system rate R=3 over AWGN channel at  $\mathcal{E}_b/N_0 = 4.76\text{dB}$ .

based on the prediction of EXIT charts, which is fairly close to the simulation results 7.3 dB when the BER curve reach  $10^{-5}$  in Fig. 3.8.

### EXIT Chart for Shaped Systems with LDPC Code

When constellation shaping is combined with LDPC codes, the generating of EXIT curve is slightly different with the case of uniform system. The shaping encoder (or decoder) is absorbed into the variable-node encoder (or decoder), and thus the contribution of the shaping code is taken into account by the VND curve. In essence, it is as if the system is using a larger modulation with high dimensionality that combines shaping with conventional modulation. The detector characteristic can then be found in a manner similar to the uniform case, i.e., by using a Monte Carlo approach. Whereas generating the detector characteristic for the uniform case involves independently modulating symbols and measuring the mutual information between the modulator input and demodulator output, the characteristic of the shaped modulation must be generated with the shaping taken into account. In reference to Fig. 3.2, a bit sequence  $\mathbf{v}$  is randomly generated and passed through the pictured processing to produce the modulated sequence  $\mathbf{x}$ . The received noisy symbols  $\mathbf{y}$  are processed to produce

the extrinsic information  $L_e(\mathbf{v})$ , and the detector characteristic is found by computing the mutual information between  $\mathbf{v}$  and  $L_e(\mathbf{v})$ . The EXIT charts of the  $R = 3$  shaped system with LDPC codes are shown in Fig. 4.9, from which we can see that the curve of VND just touches the curve of CND and the recorded SNR 4.76 dB is the threshold value. Compared with 4.96 dB when the BER is  $10^{-5}$  in Fig. 3.8, the EXIT charts provide valid prediction for the system performance without the need of extensive simulations.

## 4.2 LDPC Codes Optimization

The generated EXIT curves not only predict the threshold value but also provide the guidance for the LDPC code design. Based on the conclusion from [53], the shape of VND and CND curves should be matched to provide better error correcting capability. The DVB-S2 standard LDPC code  $(k_c, n_c)$  is used as the basis for the LDPC code degree distribution optimization. The parity-check matrix  $\mathbf{H}$  of such a code contains an arbitrary matrix for the first  $k_c$  columns and a dual-diagonal matrix as its last  $n_c - k_c$  columns. The code is further limited to be *check regular*, and its check-node degree  $d_c$  is kept the same as the value of the DVB-S2 standardized code of the same rate and length. For instance, the standardized rate  $R_c = 3/5$  LDPC code corresponds to the parity-check matrix with dimension  $(25\,920 \times 64\,800)$  and constant row weight  $d_c = 11$ , and the comparable optimized code is constrained to have the same check-node degree.

The notations from [53] are adopted and we assume there are  $D$  different variable-node degrees  $d_{v,i}$ ,  $i = 1, \dots, D$ . The fraction of nodes having degree  $d_{v,i}$  is denoted  $a_i$ , while the fraction of edges incident to variable nodes of degree  $d_{v,i}$  is denoted  $b_i$ . Note that  $a_i$  and  $b_i$  need satisfy the summation constraint, which is  $\sum_i a_i = \sum_i b_i = 1$ . Since the total degree of the variable node and check node is the same:

$$\sum_{i=1}^D (n \cdot a_i) \cdot d_{v,i} = (n - k) \cdot d_c, \quad (4.16)$$

and the edge perspective degree distribution  $b_i$  could be related to the node perspective degree distribution  $a_i$  by:

$$b_i = \frac{na_i \cdot d_{v,i}}{(n - k)d_c} = \frac{a_i d_{v,i}}{(1 - R)d_c}. \quad (4.17)$$

The designing of LDPC code involves specifying the variable node degree  $d_{v,i}$ , ( $i = 1, \dots, D$ ) and the corresponding degree distribution. The EXIT curve of a mixture codes is an average of the EXIT curves with different variable node degree, and the averaging is based on the edge fraction  $b_i$  since the extrinsic information passed along edges between variable nodes and check nodes.

### 4.2.1 Codes Designed for AWGN Channel

The code optimization is processed for uniform and shaping system individually because of the different code rates. The code rate  $R_c = 3/5$  and  $R_c = 2/3$  LDPC code are considered for uniform and shaping system, respectively. According to the constraint expressed in equation (4.16) and (4.17), only 2 degree distribution could be adjusted. The number of distinct variable node degree is initialized to be 3, i.e.,  $D = 3$ , which is the same number allowed in the DVB-S2 standard. Then we increase  $D$  by one, and optimize  $D = 4$  LDPC codes with the same approach. Due to the dual-diagonal part of the  $\mathbf{H}$  matrix, the minimum<sup>1</sup> variable-node degree is  $d_{v,1} = 2$ . This implies that the fraction of degree-2 nodes is (at least<sup>2</sup>)  $a_1 = (n - k)/n$ ; i.e., the fraction of columns corresponding to the dual-diagonal part of  $\mathbf{H}$ .

To constrain complexity, the maximum node degree is limited to be no greater than 25, in other words,  $d_{v,D} \leq 25$ . When  $D = 3$  is considered, the key step in the optimization is to determine the integer values of  $d_{v,2}$  and  $d_{v,3}$ . These are related to the constraints on the code rate and total number of 1s in equation (4.16) and (4.17), thus  $d_{v,3}$  can be determined once  $d_{v,2}$  is set. Furthermore, once  $d_{v,2}$  and  $d_{v,3}$  are determined, the corresponding degree distributions,  $a_2$  and  $a_3$ , can be found by using the check-node degree  $d_c$ . The optimization is therefore over just one parameter. On the other hand, the optimization is over two parameters when  $D = 4$ ; two of the values in the 3-tuple  $\{d_{v,2}, d_{v,3}, d_{v,4}\}$  must first be fixed before the remaining value can be determined.

---

<sup>1</sup>In DVB-S2, there is a single degree-1 variable node corresponding to the last column of the  $\mathbf{H}$  matrix. While our code design maintains this degree-1 node, we do not explicitly list the fraction of nodes connected to degree-1 nodes since it will approach zero for large  $n_c$ .

<sup>2</sup>We found no advantage to using  $a_1 > (n - k)/n$ .

### uniform system

Firstly we tried to design a rate  $3/5$  ( $R = 3/5$ ) LDPC code with the check node degree  $d_c = 11$ , which is the same as the value of DVB-S2 standard code. The exact same rate standard code is used in the  $R = 3$  uniform system introduced in subsection 3.3.2. The design approach is to find a VND EXIT curve that lies above the CND curve and minimize the distance between these two curves. One code found to be better is with variable node parameters listed below:

$$\begin{aligned} d_{v_1} &= 2 & a_1 &= 0.40 & b_1 &= 0.182 \\ d_{v_2} &= 4 & a_2 &= 0.52 & b_2 &= 0.473 \\ d_{v_3} &= 19 & a_3 &= 0.08 & b_3 &= 0.345 \end{aligned}$$

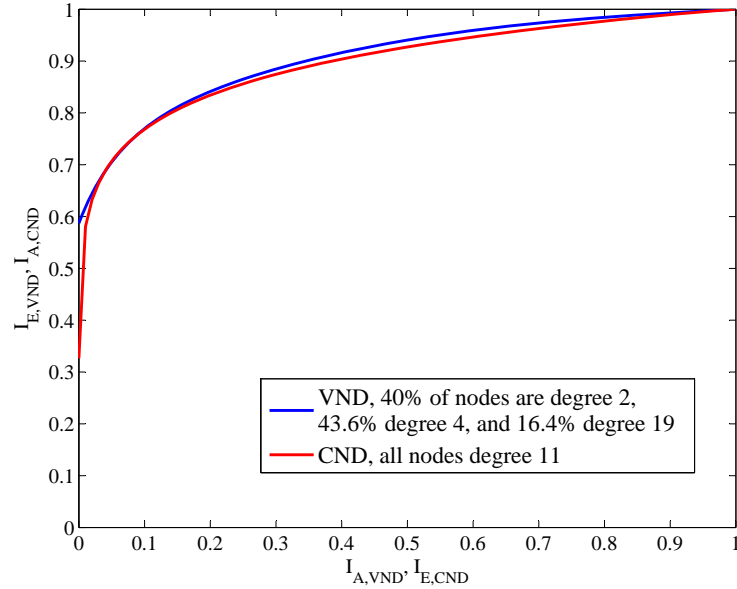


Figure 4.10: Curve fit for an LDPC code at code rate  $R = 3/5$  with uniform 32-APSK constellation,  $E_b/N_0 = 4.5\text{dB}$

The process was repeated for  $D = 4$ . Because the optimization for  $D = 4$  was over two parameters, there were several degree distributions offering low EXIT-chart thresholds. A total of 50 candidates were identified, and the realized code for each was simulated. The code offering the best simulated performance has the following degree distribution:

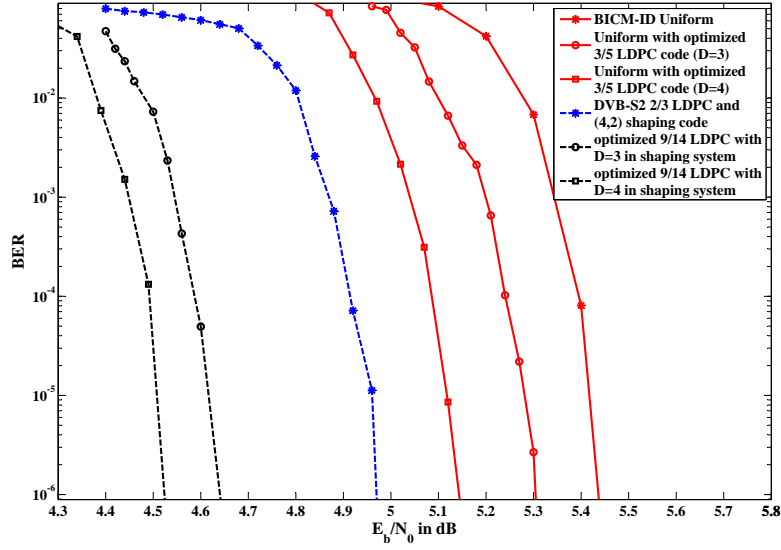


Figure 4.11: Bit-error rate of 32-APSK in AWGN at rate  $R = 3$  bits/symbol with a length  $n_c = 64800$  LDPC code. The solid lines are for the uniform system; the rightmost curve shows the performance of uniform modulation with a standard DVB-S2 code, while the two curve to its left show the performance of uniform modulation with optimized  $D = 3$  and  $D = 4$  LDPC codes, respectively. The dashed lines are for the shaped system; with the left two curves show the performance of shaped modulation with optimized  $D = 3$  and  $D = 4$  LDPC codes, and the third curve from left shows the performance of shaped modulation with a standard DVB-S2 code

$$\begin{aligned}
 d_{v,1} &= 2 & a_1 &= 0.40 & b_1 &= 0.182 \\
 d_{v,2} &= 3 & a_2 &= 0.10 & b_2 &= 0.066 \\
 d_{v,3} &= 4 & a_3 &= 0.44 & b_3 &= 0.402 \\
 d_{v,3} &= 25 & a_3 &= 0.06 & b_3 &= 0.351
 \end{aligned}$$

The uniform 32-APSK system was simulated using a length  $n_c = 64,800$  code with these two degree distributions, and the resulting BER curves with BICM-ID reception are shown in Fig. 4.11. The second and third curves from right are generated by using optimized rate  $R_c = 3/5$  LDPC code with  $D = 3$  and  $D = 4$ . These codes indicate the gain of 0.14 dB and 0.30 dB relative to the DVB-S2 standard code at a BER equal to  $10^{-5}$ . The LDPC code with one more variable-node degree ( $D = 4$ ) achieves better performance.

### Shaping system

When shaping code and rate  $2/3$  LDPC code are used, EXIT charts can also guide the process of designing better code. Instead of matching the VND and CND curve, we compare

the threshold value of codes with different degree distributions, which is the recorded SNR when the combined detector/VND curve just touches the CND curve. By considering the same decoding complexity, the codeword length and check node degree are required the same as those defined in DVB-S2 standard. Here is one code example with code length 64800 and  $d_v = 10$ , which has the threshold value of 4.71dB and the threshold value for standard same rate LDPC code is 4.73 dB.

$$\begin{aligned} d_{v_1} &= 2 & a_1 &= 0.333 & b_1 &= 0.200 \\ d_{v_2} &= 3 & a_2 &= 0.606 & b_2 &= 0.546 \\ d_{v_3} &= 14 & a_3 &= 0.061 & b_3 &= 0.254 \end{aligned}$$

Because of the limited LDPC code rate  $R_c$  in DVB-S2 standard, the  $p_0$  of the combined shaping code (with rate  $R_s$ ) may not be the optimum value identified in Section 2.3. For instance, if the overall rate  $R = 3$  is required, then  $R_c = 2/3$  and  $R_s = 1/2$  based on equation (2.26). But the value of  $p_0$  is 0.8125 when using (4, 2) shaping code, which differs from the optimal value of  $p_0 = 0.716$  when system rate is 3. A more optimal but same decoding complexity shaping code (2, 3) code gives a  $p_0 = 0.75$ , but requires LDPC code rate  $R_c = 9/14$  to achieve an overall rate of  $R = 3$  bits/symbol. There is no rate  $R_c = 9/14$  code defined in DVB-S2 standard. However, a new code can be designed for this rate by using EXIT-chart based techniques. As with the rate-2/3 standardized codes, the check node degree of rate  $R_c = 9/14$  code is set to be  $d_c = 10$ . For  $D = 3$ , the code found to have the best performance has variable-node degree distribution:

$$\begin{aligned} d_{v,1} &= 2 & a_1 &= 0.357 & b_1 &= 0.200 \\ d_{v,2} &= 3 & a_2 &= 0.558 & b_2 &= 0.469 \\ d_{v,3} &= 14 & a_3 &= 0.085 & b_3 &= 0.331 \end{aligned}$$

For  $D = 4$ , the code found to have the best performance has variable-node degree distribution:

$$\begin{aligned} d_{v,1} &= 2 & a_1 &= 0.357 & b_1 &= 0.200 \\ d_{v,2} &= 3 & a_2 &= 0.548 & b_2 &= 0.461 \\ d_{v,3} &= 5 & a_3 &= 0.002 & b_3 &= 0.002 \\ d_{v,4} &= 13 & a_4 &= 0.093 & b_4 &= 0.337 \end{aligned}$$

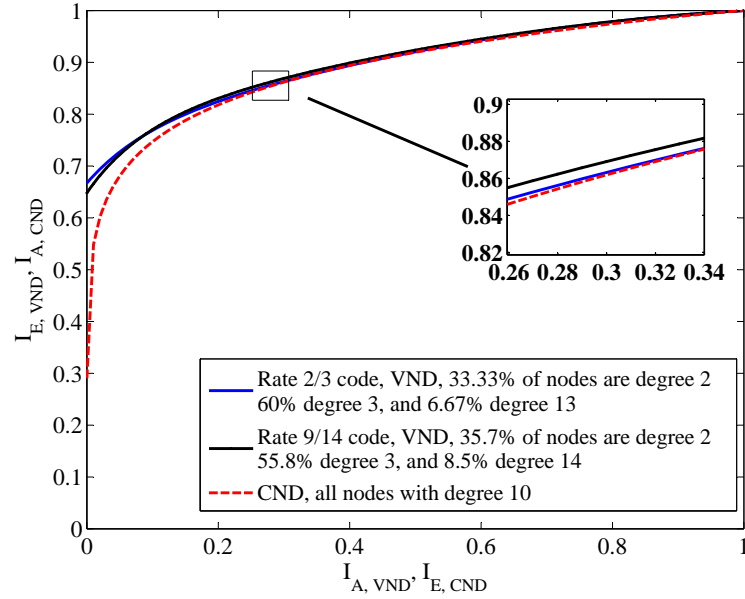


Figure 4.12: EXIT chart for BICM-ID using M=32 APSK and overall system rate  $R=3$  over AWGN channel at  $\mathcal{E}_b/N_0 = 4.73\text{dB}$ .

The EXIT chart for the optimized rate 9/14 optimized LDPC code and DVB-S2 standard code with rate 2/3 are shown in Fig. 4.12, the one for the optimized rate 2/3 LDPC code is not shown because of the space limitation. As we can see from the figure, the combined VND/detector curve of rate 2/3 standard code just lies above and touches the CND curve when  $\text{SNR} = 4.73\text{ dB}$ . But larger space is existed between the optimized rate 9/14 LDPC code and the CND curve, which means it has smaller threshold value and the simulation results in Fig. 4.11 validates its better performance when combining with (2,3) shaping code.

The designed LDPC codes are tested in the actual system simulation with 32-APSK modulation. Fig. 4.11 shows the bit-error rate curves over an AWGN channel after 100 iterations. All systems considered for simulations are with the same system rate  $R = 3$  bits/symbol. The shaped system without optimized LDPC codes provides a gain of about 0.6 dB over the uniform system at a bit-error rate of  $10^{-5}$ . Both systems implemented with optimized LDPC code achieve better performance, for uniform system the gain is about 0.1 dB while for shaping system the gain is around 0.18 dB. By comparing the performance between the system with optimized LDPC code, the shaping gain is about 0.65 dB. The gain

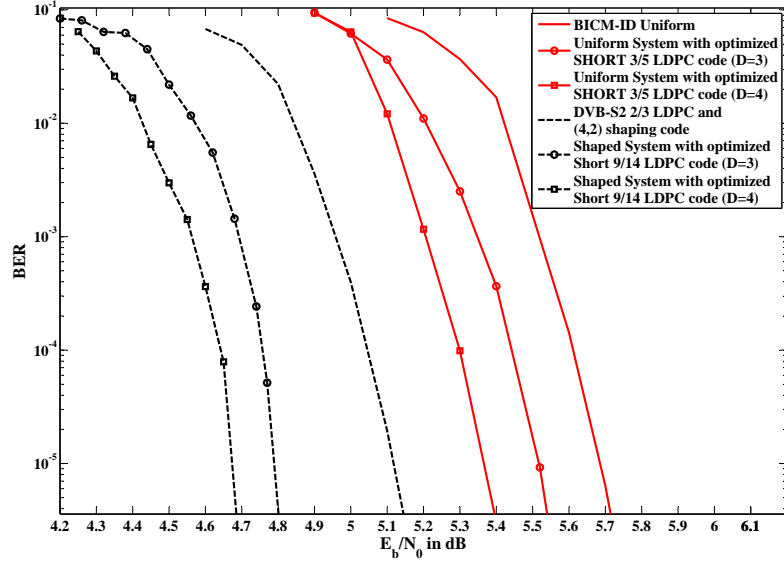


Figure 4.13: Bit-error rate of 32-APSK in AWGN at rate  $R = 3$  bits/symbol with short length LDPC codes ( $n_c = 16\,200$ ). The solid lines are for the uniform system; from right to left, the curves show uniform system using  $R_c = 3/5$  DVB-S2 standard code, uniform system with optimized code with  $D = 3$  and  $D = 4$ . The dashed lines are for the shaped system; from left to right, the curves show performance using optimized  $R_c = 9/14$  LDPC code with  $D = 4$  and  $D = 3$  combined with  $(3, 2)$  shaping code, the third curve from left shows the performance of standard  $R_c = 2/3$  code with  $(4, 2)$  shaping code

is higher if comparing the best shaping system with the uniform system without optimized LDPC code. This shaping gain is actually higher than the 0.265 dB gain predicted by the information-theoretical analysis (see Table 2.1). This additional gain can be attributed to the use of the shaping code and the inclusion of its decoder in the overall decoding loop. Essentially, the shaping code adds some additional coding gain over the gain of the LDPC code.

The BER performance of the two  $R_c = 9/14$  length  $n_c = 64\,800$  LDPC codes combined with a  $R_s = 2/3$  shaping code are shown in Fig. 4.11. The leftmost curve is generated by  $D = 4$  LDPC code, and the curve to its right is generated by using  $D = 3$  LDPC code. The results indicate a gain at BER  $10^{-5}$  of 0.44 dB compared to the shaped system that uses the DVB-S2 standard rate  $R_c = 2/3$  LDPC code along with a  $(4, 2)$  shaping code, about 0.1 dB of the gain can be attributed to better designed LDPC code with more variable node degree.

While the results in Fig. 4.11 are for long ( $n_c = 64\,800$ ) LDPC codes, Fig. 4.13 compares the bit-error performance of shorter length  $n_c = 16\,200$  LDPC codes. In particular, the three



solid curves on the right show the performance of uniform systems using short  $R_c = 3/5$  LDPC codes, with the second and third curve from right corresponding to the optimized LDPC code with  $D = 3$  and  $D = 4$ , respectively. The three dashed curves on the left correspond to shaped systems. The third curve from left is generated using the  $R_c = 2/3$  LDPC code from DVB-S2 standard combined with a  $(n_s, k_s) = (4, 2)$  shaping code. The first and second curve from left incorporate the short  $R_c = 9/14$  LDPC code optimized for shaping system with  $D = 4$  and  $D = 3$ , respectively. At a BER of  $10^{-5}$ , optimizing the code for the uniform system provides a gain over the DVB-S2 standardized code of 0.37 dB, while optimizing the code for the shaped system provides a gain of 0.43 dB.

### 4.2.2 Codes Designed for Fading Channel

The code optimization results discussed above are all generated over AWGN channel. The same optimization approach can be used for fading channel, but before that it is interesting to see the results when the optimized codes over AWGN channel are applied directly in the fading channel. The term ‘AWGN optimized’ is referred to the optimized code derived over AWGN channel, ‘fading optimized’ is referred to the optimized code derived over ergodic rayleigh fading channel.

The performance of the DVB-S2 standard LDPC codes and AWGN optimized codes over fading channel are shown in Fig. 4.14, which includes the results of both the uniform system and shaping system. The two curves at the right side interpret the performance of uniform system with BICM and BICM-ID receiver, both with standard rate  $R_c = 3/5$  LDPC code. The third curve is plotted with AWGN optimized  $R_c = 3/5$  LDPC code in uniform BICM-ID system. The rightmost dashed curve in Fig. 4.14 shows the performance with AWGN optimized  $R_c = 9/14$  LDPC code combined with  $(3, 2)$  shaping code. The second and third dashed curve from left show the performance of AWGN optimized  $R_c = 2/3$  LDPC code and standard  $R_c = 2/3$  code with  $(4, 2)$  shaping code over Rayleigh fading channel. The optimized code designed for AWGN channel could also provide better performance in fading channel. This result is reasonable since the fading channel model mentioned earlier could be treated as an AWGN channel with received signal-to-noise ratio  $|h|^2 \text{SNR}$ , where  $h$  is the

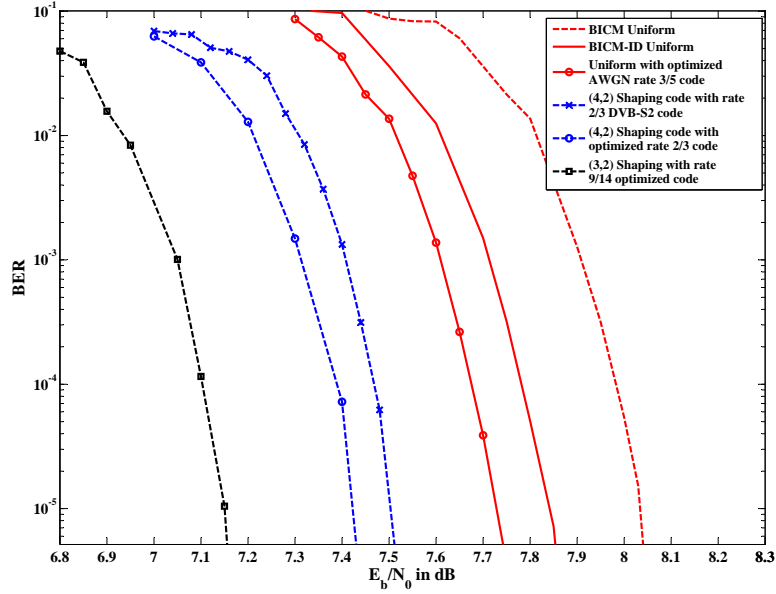


Figure 4.14: Bit-error rate of 32-APSK in ergodic Rayleigh fading channel at rate  $R = 3$  bits/symbol with a length  $n_c = 64800$  LDPC code. The three curves on the right side are for the uniform system, the three curves on the left side are for the shaped systems.

channel gain coefficient. The benefit of using AWGN optimized LDPC code is about 0.1 dB for the uniform system with BICM-ID receiver; the optimized  $R_c = 9/14$  code could provide about 0.45 dB performance gain when combining with  $(3, 2)$  shaping code compared with the standard  $R_c = 2/3$  LDPC code with  $(4, 2)$  shaping code.

On the other side, the optimized LDPC code could be found based on the EXIT chart generated over fading channel. The fading optimized LDPC code should be more appropriate for the system over fading channel than the previous generated AWGN optimized LDPC code. Several new LDPC codes are found and their simulation results are shown in Fig. 4.15. The solid curves from right to left show the performance of uniform systems. The leftmost two curves stand for the BICM and BICM-ID system with standard rate  $3/5$  LDPC code. The other two solid curves are the uniform systems with  $R_c = 3/5$  AWGN optimized and fading optimized code. The optimized  $R_c = 3/5$  LDPC code based on fading channel performs extremely well in uniform system with BICM-ID receiver. The optimized code is a eIRA code with  $d_c = 11$  and has 3 different variable node degree, i.e.,  $D = 3$ , and the degree distributions are:

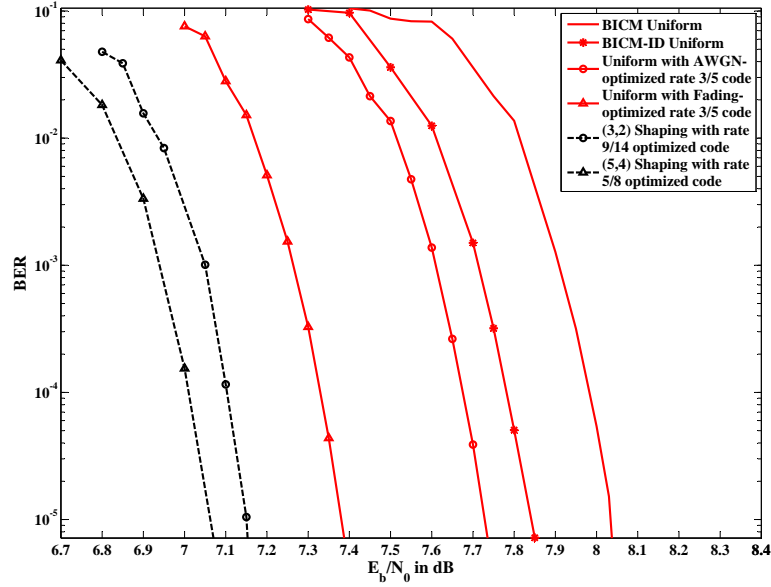


Figure 4.15: Bit-error rate of 32-APSK in ergodic Rayleigh fading channel at rate  $R = 3$  bits/symbol with a length  $n_c = 64800$  LDPC code. The solid lines are for the uniform system, The dashed lines are for the shaped system.

$$\begin{aligned}
 d_{v,1} &= 2 & a_1 &= 0.400 & b_1 &= 0.182 \\
 d_{v,2} &= 3 & a_2 &= 0.488 & b_2 &= 0.332 \\
 d_{v,3} &= 19 & a_3 &= 0.112 & b_3 &= 0.486
 \end{aligned}$$

The performance gain of using fading optimized code in uniform system is about 0.45 dB compared with the standard LDPC code. The fading optimized code adopted in uniform system even provides better performance than the shaping system with standard  $R_c = 2/3$  LDPC code. The reason might be the vulnerability of the shaping code in fading channels. Based on the optimum  $p_0$  value 0.688, a better shaping code (5,4) with  $p_0 = 0.6875$  is chosen to outperform the original shaping systems. The code rate  $R_c = 5/8$  LDPC code with  $d_c = 10$  is found to combine with (5,4) shaping code so that the system rate could be  $R = 3$ . The degree distributions of the fading optimized  $R_c = 5/8$  code is:

$$\begin{aligned}
 d_{v,1} &= 2 & a_1 &= 0.375 & b_1 &= 0.200 \\
 d_{v,2} &= 3 & a_2 &= 0.563 & b_2 &= 0.45 \\
 d_{v,3} &= 21 & a_3 &= 0.062 & b_3 &= 0.35
 \end{aligned}$$

The dashed rightmost curve in Fig. 4.15 shows performance with fading optimized  $R_c = 5/8$  LDPC code combined with (5,4) shaping code, the second dashed curve from left shows

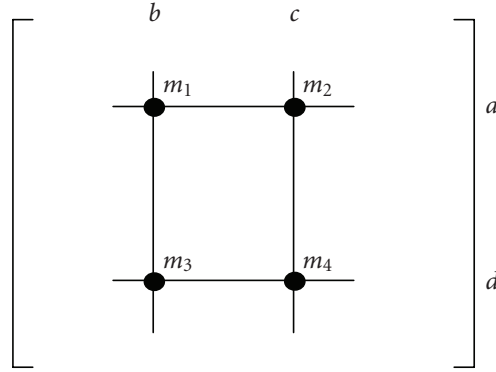


Figure 4.16: The form of a 4 cycle in a parity-check matrix

the performance of AWGN optimized  $R_c = 9/14$  LDPC code combined with  $(3, 2)$  shaping code. The shaping system with fading optimized  $R_c = 5/8$  LDPC code could provide about 0.43 dB gain compared with the shaping system with the standard  $R_c = 2/3$  LDPC code with  $(4, 2)$  shaping code. The shaping gain could be up to 0.79 dB when evaluating the performance of shaping system with optimized  $R_c = 5/8$  LDPC code and uniform system with standard  $R_c = 3/5$  LDPC code. Even using fading optimized code in both uniform and shaping system, the shaping gain is around 0.3 dB.

### 4.3 Short Cycles in LDPC Codes

When the optimized LDPC codes are found according to the methodology mentioned above, the next step is to recover the full parity-check matrices based on the derived degree distributions. Other than the pre-defined dual diagonal sub-matrix, 1s are randomly inserted in the parity check matrix. The locations of 1s in the matrix may cause the issue of short cycles.

The definition of cycles is easier to understand based on the Tanner graph of LDPC codes, which is the graphical representation of the parity check matrix. A path in the graph refers to a set of edges with the connecting vertices. A cycle is a path that the starting and ending vertices are the same. The number of edges of the cycle is called its length, which

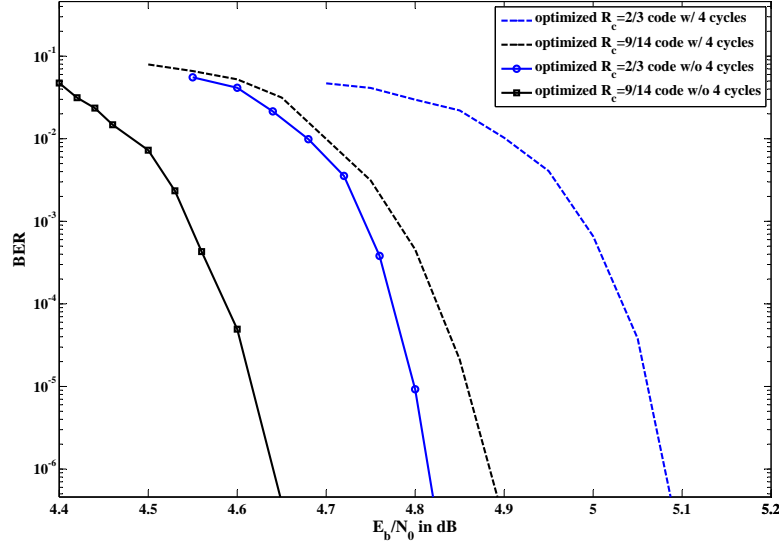


Figure 4.17: Bit-error rate of shaped 32-APSK at rate  $R = 3$  bits/symbol with a length  $n_c = 64800$  LDPC code over AWGN channel. The solid lines are results with the LDPC codes without 4 cycles, the dashed lines are for the LDPC codes with 4 cycles.

should be an even number in a Tanner graph. The existence of cycles, especially the cycles with short length, will affect the independence of the information exchanged along the edges of Tanner graph and degrade the iterative decoding process of the LDPC decoder [54, 59]. An example of cycle with length 4 is plotted in Fig. 1.6 and the form in the corresponding matrix is shown in Fig. 4.16.

The matrix that contains four cycles has two 1s in the same row  $a$ , and the columns they are belong to have another two 1s in the same row other than  $a$ . Due to the impact of preventing the converging of the decoding algorithms, several ways are derived in [16, 60, 61, 62, 63] to construct LDPC codes with large girth, which is the length of the shortest cycle in the Tanner graph. However, the construction methods derived not always generate the LDPC code with desired code rate or specified structure. One of the popular approaches is the so-called *progressive-edge-growth* (PEG) algorithm proposed by Hu et al. [16]. The PEG algorithm builds up the parity-check matrix, or equivalently a Tanner graph on an edge basis and by maximizing the local girth at variable nodes. Assume the constructing of the first  $j - 1$  variable nodes and their connecting edges have been established, the girth of current graph is  $g_t$ . The edge set of variable node  $j$  is selected such that the girth of the current graph will not be reduced. The PEG algorithm is flexible enough to construct LDPC codes

of arbitrary length and rate, but the complexity due to the exhaustively searching for the optimum edge set will rapidly grow with the increasing of codeword length, especially for the LDPC codes with codeword length 64 800.

The methodology used in this dissertation is based on the software provided by Radford M. Neal [64], which randomly inserts 1s based on the node perspective degree distributions and tries to eliminate the cycles with length 4. The matrix generated by Neal's software is combined with the dual diagonal matrix to be the parity check matrix used for the simulations. Fig. 4.17 contain the comparison results between LDPC codes with and without 4 cycles. The system scheme used is the BICM-ID with shaped 32-APSK modulation. The solid curves are plotted with optimized LDPC codes that are tried to eliminate the 4 cycles. While the dashed curves are the results with the parity-check matrices randomly generated without any procedure of eliminating the 4 cycles. The performance degradation caused by the 4 cycles could be around 0.27 dB according to the curves in Fig. 4.17. The performance gain achieved by constellation shaping or LDPC code optimization could be mitigated if ignoring the construction of the LDPC codes parity-check matrix.

## 4.4 Conclusion

The EXIT charts can be used to illuminate the convergence properties of the BICM-ID systems with or without constellation shaping strategy. The idea of code optimization based on EXIT charts are presented in this chapter. The numerical simulation results validate this approach. The newly designed LDPC used in uniform system can provide about 0.12 dB performance gain when comparing with the same rate uniform system with DVB-S2 standard LDPC code. After the combination of new rate LDPC code with proper shaping code, the shaping system described in chapter 3 can provide additional 0.8 dB gain compared with the uniform system in AWGN channel at the same transmission rate of 3 bits per symbol. By including both optimized LDPC code in uniform and shaped system, around 0.67 dB performance gain can be observed from the simulation BER curves. The benefit of the optimized code will be reduced if neglecting the issue of short cycles in the construction process of parity-check matrix.

## Chapter 5

# Constellation Optimization

In previous chapters, we have presented research on constellation shaping and related optimization strategies. In chapter 2, the input constellation probability distributions are optimized by maximizing the mutual information with the non-linear shaping code. In chapter 3, the optimized LDPC codes are found for both shaped and unshaped constellations through an EXIT chart analysis. Another interesting problem that needs further attention is constellation optimization, especially when combined with the constellation shaping technique. This chapter discusses two aspects of constellation optimization, optimizing the symbol-labeling map, and optimizing the placement of symbols in N-dimensional space. The focus is on the APSK constellation. All the possible optimization strategies are combined together at the end of this chapter.

Throughout this chapter, we use  $\mathcal{X}$  to represent the set of constellation symbols used for transmission. Each element of  $\mathcal{X}$  is an  $N$ -dimension symbol  $x_i, i = 0, \dots, M - 1$ , where  $M$  is the cardinality of the signal set, i.e.,  $|\mathcal{X}| = M = 2^m$ . The input alphabet is defined by using an  $M \times N$  matrix  $\mathbb{X} = [\mathbf{x}_0, \dots, \mathbf{x}_{M-1}]^T$ , each row of which contains one element of  $\mathcal{X}$ .

For the given input alphabet  $\mathbb{X}$ , the input distribution of the symbols is denoted by the probability mass function (pmf)  $P_X(\mathbf{x})$ , which is the probability of selecting the symbol  $\mathbf{x}$ . A matrix  $\mathbb{P}$  contains the ordered probabilities of the symbols, i.e.,  $\mathbb{P} = [P_X(\mathbf{x}_0), \dots, P_X(\mathbf{x}_{M-1})]$ . Based on the definition of pmf, the summation of the input probability of each symbol should be 1.

The labeling of  $\mathbb{X}$ , which is the assignment of the length  $m$  bit sequences to the symbols

of a  $M = 2^m$ -ary constellation, is defined by an  $M \times m$  matrix  $\mathbb{L}$ . Both the notations “mapping” and “labeling” here refer to the assignment of  $m$  bits to a symbol in the  $M$ -ary constellation. Each row of the matrix  $\mathbb{L}$  corresponds to the labeling of one input symbol  $\mathbf{x}_i$ ,  $\mathbb{L} = [(\mathbf{t}_0), \dots, (\mathbf{t}_{M-1})]^T$ , where  $(\mathbf{t}_i) = [t_{i,0}, t_{i,1}, \dots, t_{i,m-1}] \in \{0, 1\}^m$  and only the binary labeling is considered in this chapter.

A constellation is defined by the three matrices listed above, i.e., an input alphabet using a given mapping rule and input distribution. The shaping technology mentioned in chapter 2 and 3 already considered the input distribution  $\mathbb{P}$ . The optimization of the location of the alphabet  $\mathbb{X}$  and the labeling  $\mathbb{L}$  of the constellation will be discussed next.

## 5.1 Mapping Optimization Strategy

There are  $M!$  possible mapping rules for a  $M$ -ary constellation. Different mappings are selected for a variety of applications. The most popular one is Gray mapping proposed by Frank Gray to reduce the error rate in the pulse code communication system [65]. The binary labels of the two adjacent symbols differ in only one bit and Gray mapping is widely used in bit-interleaved coded modulation [34]. Another widely used mapping is set partitioning mapping (SP), which was introduced by Ungerboeck in 1982 for increasing the minimum *Euclidean distance* for the trellis coded modulation system [66]. Fig. 5.1(a) and 5.1(b) show the two mapping rules for 8-PSK modulation. In addition, the modified set partitioning (MSP) mapping from [67] and the maximum squared Euclidean weight (MSEW) mapping from [68] are also shown in Fig. 5.1.

The Euclidean distance is the direct distance between two constellation points, which can be calculated according to the ordinates of the two symbols with the *Pythagorean formula*. It could be used to characterize and analyze mappings according to the derived bound of the error probability [69].

The probability of error is defined as the probability that the estimated message  $\hat{x}$  is not the message that was transmitted. The optimum detector determines  $\hat{x}$  to minimize the error probability. The *Maximum a Posteriori* (MAP) detector is the detector that selects the estimated  $x_i$  that maximizes the a posteriori probability  $p_{\mathbf{x}|\mathbf{y}}(x_i|y)$  given a received  $y$ .



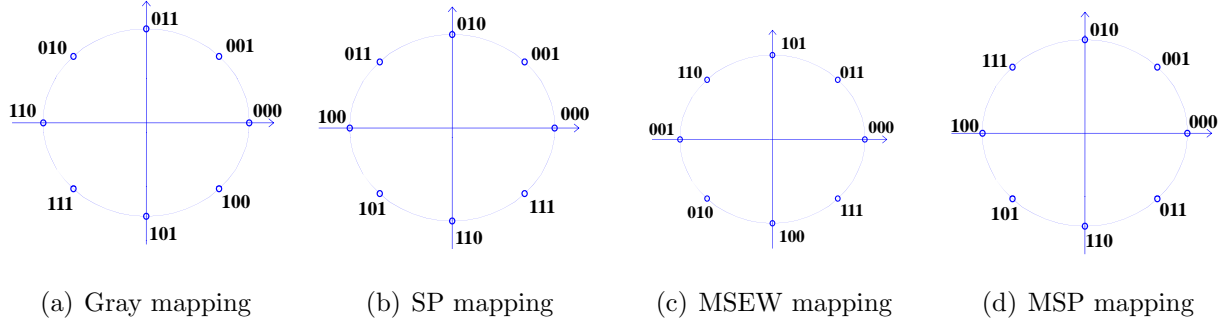


Figure 5.1: Four mappings of 8-PSK constellation

Then the decision rule of MAP detector could be written in terms the channel transition probability and the input probability distribution  $p(\mathbf{x})$ .

$$\hat{x} \rightarrow x_i \quad \text{if} \quad p_{\mathbf{y}|\mathbf{x}}(y|x_i) \cdot p_{\mathbf{x}}(x_i) \geq p_{\mathbf{y}|\mathbf{x}}(y|x_j) \cdot p_{\mathbf{x}}(x_j), \quad \forall j \neq i \quad (5.1)$$

For the AWGN channel and based on its transition probability, the MAP decision rule becomes:

$$\hat{x} \rightarrow x_i \quad \text{if} \quad \exp\left(-\frac{1}{N_0}\|y - x_i\|^2\right) \cdot p_{\mathbf{x}}(x_i) \geq \exp\left(-\frac{1}{N_0}\|y - x_j\|^2\right) \cdot p_{\mathbf{x}}(x_j), \quad \forall j \neq i \quad (5.2)$$

If the channel input symbols are with equal probability and equation (5.2) is rewritten in the log-domain, the MAP detection rule becomes the AWGN ML detection rule: For the AWGN channel and based on its transition probability, the MAP decision rule becomes:

$$\hat{x} \rightarrow x_i \quad \text{if} \quad \|y - x_i\|^2 \leq \|y - x_j\|^2 \quad \forall j \neq i \quad (5.3)$$

Either the MAP or ML detector will estimate the channel output  $y$  to be one of the  $M$  possible transmitted symbols. The complex plane could be partitioned into  $M$  regions, each of which corresponds to one of the  $M$  possible decisions. The ML detector rule in (5.3) indicates that the decision of  $\hat{x}$  is determined by choosing the symbol  $x_i$  that is closest to the received channel output  $y$ . In the coded modulation system, bit-wise decisions are more relevant than the symbol wise decisions since the decoder receives the soft bit information from demodulator. The bit wise decision region is the summation of the decision regions related to the symbols that has the same bit value. Fig. 5.2 illustrates the bit-wise decision

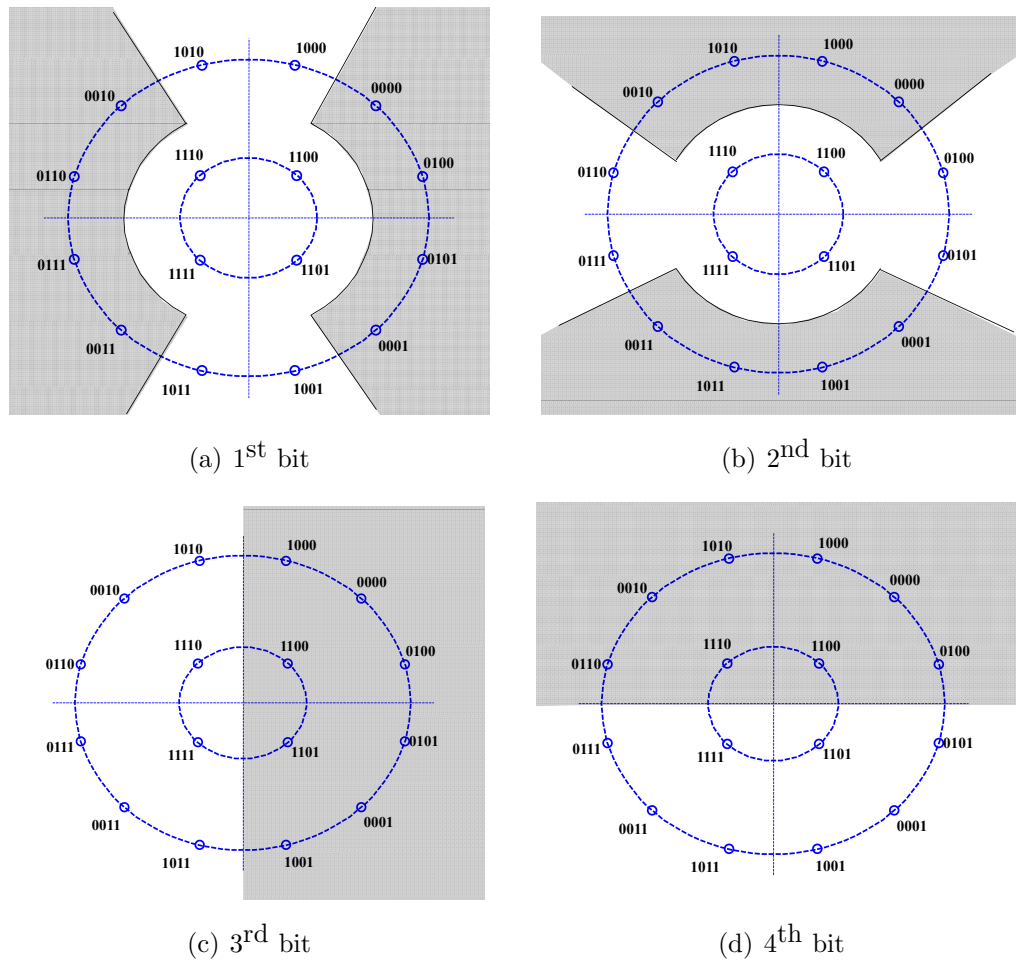


Figure 5.2: 16-APSK with mapping defined in DVB-S2 standard. Decision regions for bit  $m = 1, 2, 3, 4$  with no a priori information. Shaded region is the decision region where bit  $m$  is 0

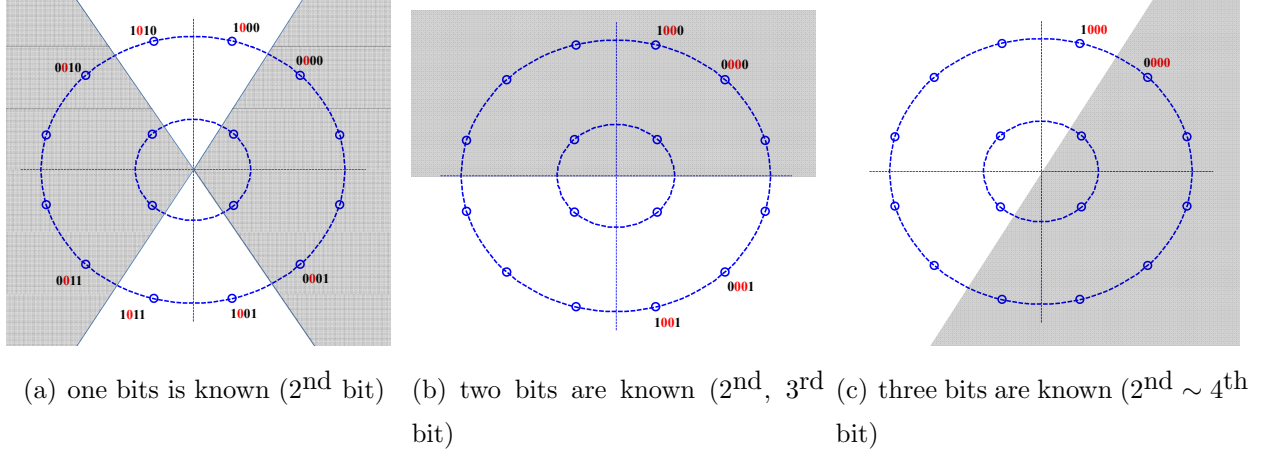


Figure 5.3: The 16-APSK symbol with label (0000) is transmitted, decision regions for the first bit equals to 0 when 1, 2, 3 bits are known, respectively. The labeling is from DVB-S2 standard.

regions of 16-APSK modulation with AWGN ML detector rule. The ML detector determines that the  $m$ -th bit is 0 when the received message located on the shaded decision region. A larger decision region may provide better protection for the corresponding bit, for example, the first bit of the 16-APSK defined in Fig. 5.2 is easier to be mistaken than that of the 4th bit.

For the demodulator with the a priori input, which corresponds to the model of BICM with iterative decoding, the decision region may be enlarged due to the knowledge of partial bits. Therefore the estimation reliability will also be improved. The bit-wise decision region with the effect of the a priori information for 16-APSK is plotted in Fig. 5.3. Each sub-figure corresponds to the decision region of the first bit at the situation of 1, 2, 3 bits are previously known, respectively. The label of the transmitted symbol is assumed to be (0000) without loss of generality. The decision region for the first bit with no a priori information could be referred to Fig. 5.2(a). The shaded area is the decision region where the first bit is estimated to be '0'. With the perfect a priori information in Fig. 5.3(c), i.e., other bits except the bit to be estimated are known, the constellation only contains two symbols that need to be evaluated, the labels of which only differ in one bit. The decision regions are similar to those with BPSK modulation.

To determine the reliability of a bit  $i, i \in 1, 2, \dots, m$ , the demodulator should consider

and average all possible a priori known bits and the bit values. Large decision regions provide high protection for the corresponding symbols or bits therefore their estimation may have high reliability. Since the constellations could be characterized by the decision regions of each bit position, the relevant Euclidean distances between the decision regions and other related parameters are used to search the suitable constellation mapping rules.

The *free squared Euclidean distance* (FED) and the FED with the ideal a priori information (FEDC) were used in [47, 70] to obtain the signal mappings for 8-PSK constellation. The FED, which could be generated by the Euclidean distance and the Hamming distance of the error code, determines the asymptotic performance for the BICM system. On the other hand, FEDC corresponds to the asymptotic performance of BICM-ID system with perfect knowledge of  $m - 1$  bits in the  $m$ -ary labeling. Yih and Geraniotis investigated the asymmetric 8-PSK to enlarge the Euclidean distances and their proposed scheme can improve the performance of a BICM-ID system [71]. A new symbol mapper called the maximum squared Euclidean weight (MSEW) mapping are proposed in [68] based on the asymptotic performance at high SNR. More recently, the Euclidean distance spectrum (EDS) is defined in [72], which is the cumulated Hamming distance between bit labels at a specific Euclidean distance. The constellation properties could be characterized by EDS and the derived distance criteria is used to design the mapping rule for BICM and BICM-ID systems [73].

The studies mentioned above are based on the Euclidean distances and related to the asymptotic performance or the derived error bounds. Since the approximation assumes the high SNR range, it may not be the suitable criteria to design signal mapping for systems operating at a practical range of SNR level or small number of iterations. Other parameters besides the Euclidean distance will be discussed below to characterize the influence of constellation mappings to the system performance. The mapping rules including Gray, set partitioning (SP), modified set partitioning (MSP) and maximum squared Euclidean weight (MSEW) are investigated in [67, 68, 74] for the 8-PSK constellation. The distance parameters are provided in [74] and SP mapping was recommended for BICM-ID 8-PSK system in [70]. However, only convolutional code is used in the aforementioned references. The results of different mapping rules combined with DVB-S2 standard LDPC code are presented in Fig. 5.4.

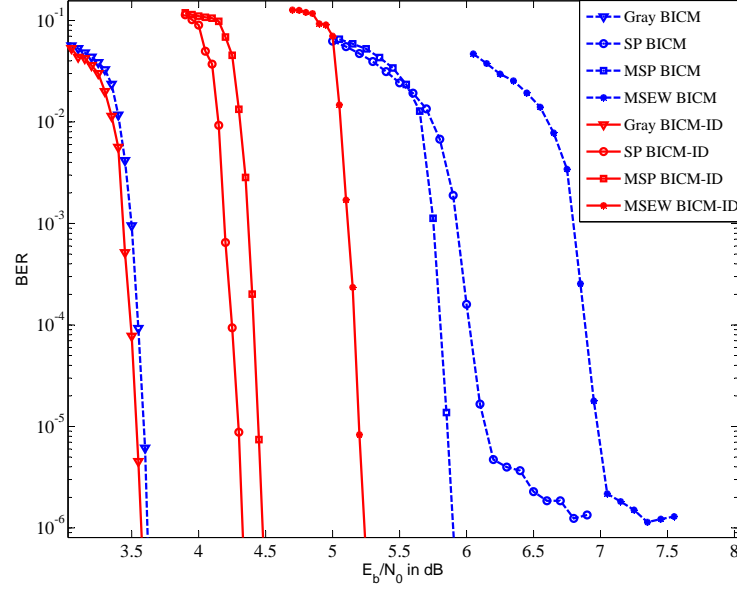


Figure 5.4: Bit-error rate of 8-PSK in AWGN at rate  $R = 2$  bits/symbol with a length  $n_c = 64800$  DVB-S2 LDPC code over AWGN channel. The dashed lines are for the uniform BICM systems; the solid lines are for the BICM-ID uniform system.

The rate of the DVB-S2 standard LDPC code in Fig. 5.4 is  $2/3$  and the maximum iteration number is set to be 100. The Gray mapping provides the best performance in the BICM system model based on the simulation results in Fig. 5.4. The same conclusion that Gray and quasi-Gray mappings are optimal for BICM systems were derived in [34, 75]. The BICM-ID system with Gray mapping performs best compared with the systems with other two mapping rules, but the gain based on the iterative decoding is marginal. The simulation results with Gray mapping agrees with the EXIT charts analysis results shown in Fig. 4.2, in which the Gray mapping demodulator mutual information transfer characteristics is almost flat when the input a priori information keeps increasing.

### 5.1.1 Constellation Labeling Optimization

For specific systems and applications, we can either use a random mapping or generate a optimum mapping based on proper algorithms. The random mapping is difficult to tell its performance unless comparing with all other mappings. The exhaustive search is intractable since a maximum of  $2^m!$  mapping sets need to be evaluated, for example, there are  $16! \approx 2 \cdot 10^{13}$  possible mappings for 16-APSK. The number is still too large after ex-

cluding those mappings with similar properties by bits mirroring or rotating. A significant system improvement can be obtained by carefully designing the labeling assignment of signal alphabet  $\mathbb{X}$  for both BICM and BICM-ID system [73, 75].

The technique to find the suitable signal mapping for 8-PSK constellation in [76, 48], for 16-QAM constellation in [67] and for 64-QAM constellation in [77] is based on the parameters related to Euclidean distance. The distance parameters *free squared Euclidean distance* (FED) and *FED conditioned on the ideal feedback* (FEDC) are reviewed in [74]. The FED and FEDC determine the asymptotic performance of the BICM and BICM-ID system, respectively. The value of these two parameters for 8-PSK are shown in Table I of [74] with four different mapping rules. The FED values are identical for all mapping rules and the FEDC value of SP and MSP mapping are also the same even though the simulation results in [74] and Fig. 5.4 indicate the opposite conclusion.

A different technique based on mutual information was introduced to explain the influence of constellation labeling on the performance of BICM-ID system [49]. Brink et al. considered BICM-ID systems with 8-PAM and 16-QAM constellations. The mutual information is employed in [74] to find the good signal mapping of various 8-ary constellations for BICM-ID systems over the AWGN channel. According to the comparison between the mapping introduced in [74] and the ones suggested in [48] based on Euclidean distance, mutual information is a more suitable technique to find the good signal labelings. The *binary switching algorithm* (BSA) algorithm presented in [73] uses different criteria to find locally optimal mappings for BICM and BICM-ID systems.

### 5.1.2 Binary Switching Algorithm and the Cost Functions

The binary switching algorithm was first introduced in [78] to search the better index assignment for quantized vector, then F. Schreckenbach et al. applied BSA to the optimization of constellation mappings [73]. The improved algorithms called adaptive BSA (ABSA) and modified ABSA are proposed in [79] and [80], respectively. Instead of brute-force searching, BSA and the improved BSA algorithms switch the labeling of constellation labels to reduce the value of provided cost function. With the proper choice of cost function, the optimized

mapping for constellations with no a priori information or ideal a priori information could be found.

The BSA is started with an initial random mapping and the optimum result is found by minimizing the chosen cost function. The first step is to calculate the cost of each constellation symbol and sort them in the descending order. Then the symbol with the highest cost in the list is selected, whose index is switched with another symbol. This candidate symbol is chosen such that the decrease of the total cost value is as large as possible. If no switch option can be found for the symbol with the highest cost, the symbol with the second-highest cost will be switched to decrease the total cost. This process repeats for every symbol in the constellation until lower total cost could not be found. Since this algorithm only finds the local optimum results, several executions with random initial mappings are necessary to find the presumed global optimum result.

The cost function needs to be defined before processing the BSA operation. The parameters related to Euclidean distance, error bounds or mutual information are the candidate cost function measures. One of the cost function is the Chernoff error bound derived in the Appendix of [72]. The parts that are only related to the Euclidean distance from the derivation of the Chernoff error bound serve as the cost function, which has the form as:

$$D_a = \frac{1}{m2^m} \sum_{i=1}^m \sum_{b=0}^1 \sum_{x \in \chi_b^i} \sum_{\hat{x} \in \chi_{\bar{b}}^i} \exp \left( -\frac{\mathcal{E}_s}{4N_0} \frac{1}{|x - \hat{x}|^2} \right) \quad (5.4)$$

for the AWGN channel and

$$D_r = \frac{1}{m2^m} \sum_{i=1}^m \sum_{b=0}^1 \sum_{x \in \chi_b^i} \sum_{\hat{x} \in \chi_{\bar{b}}^i} \exp \left( \frac{1}{|x - \hat{x}|^2} \right) \quad (5.5)$$

for the fully interleaved Rayleigh fading channel. The two equations assume a uniform input distribution, therefore this technique can only be applied to the unshaped system. The  $\mathcal{E}_s$  is the average symbol energy and  $N_0$  is the power spectrum of the white Gaussian noise. The  $\chi_b^i$  contains the subset that the  $i$ -th bit position of the labeling is  $b$ . The  $\bar{b}$  denotes the value other than  $b$  and  $\chi_{\bar{b}}^i$  denotes the neighbors of symbol  $x$ .

The inverse of the cost function in equation (5.5) is interpreted as the *harmonic mean* of the Euclidean distance in [34]. The proposed cost function in [34] is expressed in the form

Table 5.1: Values of Cost Functions related to Euclidean distance at  $\mathcal{E}_s/N_0 = 8$  dB.

	Gray	SP	MSP	MSEW
$D_r^0$	1.3047	1.5059	1.5059	1.7071
$D_r^1$	1.2357	0.8190	0.5833	0.3476
$D_a^0$	0.2788	0.3379	0.3379	0.3969
$D_a^1$	0.2661	0.1471	0.0817	0.0163

of weighted summation based on the known a priori information:

$$D_E = \lambda_0 D_0 + \lambda_1 D_1 \quad (5.6)$$

where  $D_0$  and  $D_1$  denotes the cost function under the situations of no a priori information and ideal a priori information. The corresponding weights are noted as  $\lambda_0$  and  $\lambda_1$ . The symbols in set  $\chi_b^i$  and  $\chi_b^i$  could be different with the changing of a priori information. For example, when there is no a priori information, both the two sets contains  $2^{m-1}$  symbols, while with perfect a priori information, both of the sets  $\chi_b^i$  and  $\chi_b^i$  contains one symbols.

The values of the cost function in equation (5.5) and (5.6) are given in Table 5.1 for four different mappings of 8-PSK. The notation  $D^1$  is the value with ideal a priori information and  $D^0$  is the value with no a priori information. The cost function should be minimized to achieve better performance. According to the simulation results from [73, 74] with 16-state convolutional codes in BICM-ID system, the performance with SP mapping is better than MSP mapping and MSEW mapping. However, the reasons of the error floors showed in their results are ambiguous; it could be caused by the convolutional code or the mapping rule. A more powerful LDPC code is used to avoid the error floor caused by the error correcting code. The results with LDPC codes are showed in Fig. 5.4, which indicate the similar order of performances without noticeable error floors in BICM-ID systems.

The results in Fig. 5.4 justify the cost function of  $D^0$ . The Gray mapping has the minimum value of  $D^0$  and yields the best performance. Although the cost values  $D^0$  of SP and MSP are the same, their performance difference is about 0.1 dB. Compared with the cost function with ideal a priori information  $D^1$ , the cost function  $D^0$  is more accurate to predict the performance of BICM-ID system according to the simulations results. The system with no a priori information is similar to the situation of BICM-ID system at the



first iteration. The small value of  $D^0$  provides a good start point for BICM-ID system to continue the iterative process.

Another cost function is related to the mutual information discussed in chapter 2. The symbol-wise mutual information is independent of the applied mapping rules but it could be expressed as the summation of the conditional bit-wise mutual information related to mapping. [1].

$$C = \sum_{L=0}^{m-1} I_L \quad (5.7)$$

where  $I_L$  is conditional bit-wise mutual information under the assumption that perfect knowledge of “ $L$  other bits in one symbol are known”. The value of  $I_L$  is calculated by averaging all possible situations of value of  $L$ , all bit locations in mapping and the bit value 0 or 1. The increasing of known bits also means the decreasing of the constellation size, for instance, the  $M$ -ary constellation becomes BPSK once  $m - 1$  bits are known. The conclusion that  $I_0 \leq I_1 \leq \dots I_{m-1}$  could also be drawn since the conditioning increases the mutual information.

Notice that the binary label  $\mathbf{t}_k$  of the symbol  $x_k$  is  $[t_{k,0}, t_{k,1}, \dots, t_{k,m-1},] \in \{0, 1\}^m$ . The  $m$  bits are independent due to the bit-based interleaver in the BICM system. The input constellation symbol probability is the multiplication of the probability of each bit:

$$p(x_k) = \prod_{i=1}^m p_{t_i}(t_{k,i}) \quad (5.8)$$

where  $t_i$  is the random variable representing the  $i$ th bit in the codeword. And the probability of the symbol  $x_k$  conditioned on the  $i$ th bit being  $b$  is:

$$p_{x|t_i}(x_k|b) = \frac{p(x_k)}{p_{t_i}(b)} \text{ if } x_k \in \chi_{i,b} \quad (5.9)$$

where  $\chi_{i,b}$  is the subsets of symbols whose  $i$ th bit labeling is  $b$ , i.e.,  $\chi_{i,b} = \{x_k \in \chi : t_{k,i} = b\}$ .

The bit-wise mutual information in BICM system could be expressed as:

$$I = \sum_{i=1}^m \sum_{b=0}^1 \sum_{k \in \chi_b^i} P(x_k) \int p(y|x_k) \cdot \log \frac{\sum_{k' \in \chi_b^i} p_{x|t_i}(x_{k'}|b) p(y|x_{k'})}{\sum_{k' \in \chi} p(y|x_{k'})}, \quad (5.10)$$

If all the  $m$  bits are with equal probability,  $p(x_k) = 1/M$  and  $p_{t_i} = 1/2$  for  $i = 1, 2, \dots, m$ . The derivation of the BICM mutual information and its expression form with uniform input probability distribution could be found in [73] and [81] .

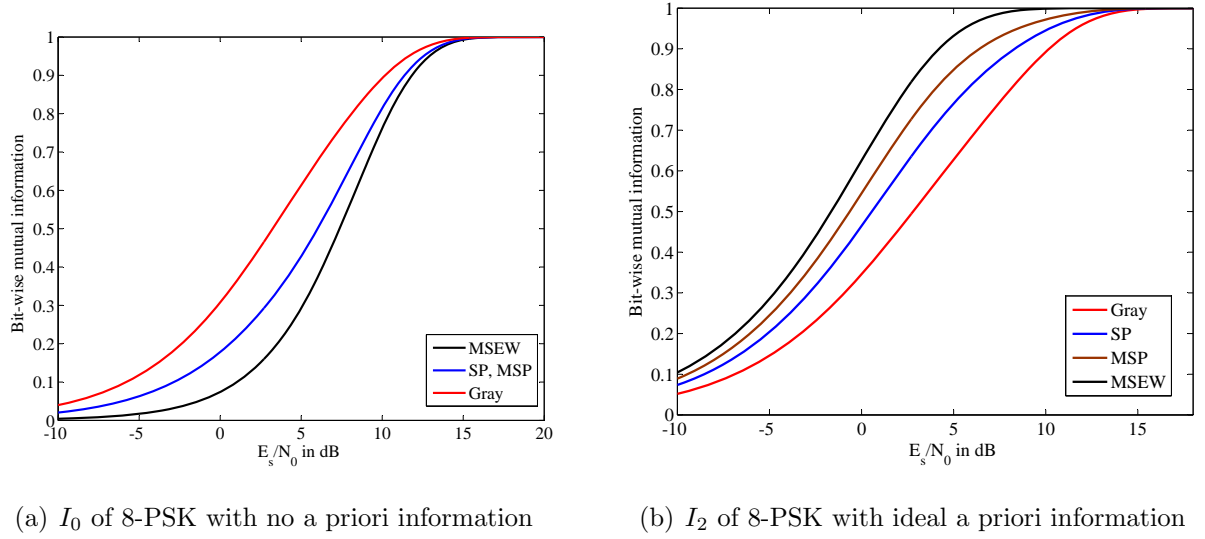


Figure 5.5: Bit-wise mutual information of 8-PSK with different mapping rules

The conditional bit-wise mutual information  $I_0$  assumes that there is no information from the other bits, which is an important parameter for BICM system with no iterative decoding. The channel capacity of BICM system is  $mI_0$  when all bits are uniformly distributed [34]. When the iteration between the decoder and the demodulator is started, more accurate information about the other bits in one single symbol will be feed back to the demodulator at the next iteration. Then the importance of  $I_0$  decreases and  $I_1, I_2, \dots, I_{m-1}$  become dominant parameters to the error performance of BICM-ID system as the iteration number increases. When the perfect a priori knowledge is available, which means the other  $m - 1$  bits are known, then only  $I_{m-1}$  dominates the performance of BICM-ID system.

The summation of the conditional bit-wise mutual information is the symbol-wise mutual information, which is a fixed value for the same constellation. Therefore, there is a tradeoff in the values of the conditional bit-wise mutual information. A larger  $I_0$  value means the other mutual information with a priori knowledge will be smaller, while a small  $I_0$  value may result in poor performance of BICM system or BICM-ID systems at the first iteration. A compromise solution is to design the mapping with  $I_0$  big enough to start the iteration while maximizing  $I_{m-1}$  to achieve good performance after large number of iterations [74].

Fig. 5.5 plots  $I_0$  and  $I_2$  for 8-PSK of four different mappings. Notice that  $I_0$  is the same for SP and MSP mapping, which is similar to the derived results based on the cost function

$D^0$  in Table 5.1. Normally the higher conditional bit-wise mutual information results in a better asymptotic error performance, but the curves of  $I_2$  behave in the opposite order compared with the bit error rate curves in Fig. 5.4. Due to the advantage of predicting system performance, both  $I_0$  and  $D^0$  are used as the criteria in the following optimizations for the mapping rule of 32-APSK.

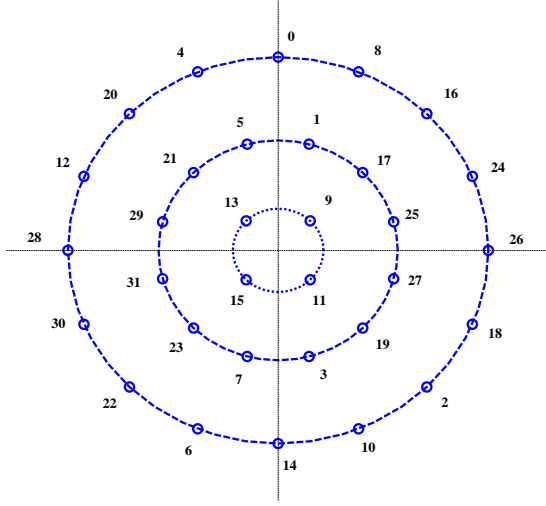
### 5.1.3 Mapping Optimization for Uniform 32-APSK

The DVB-S2 standard 32-APSK is used as the start point for the mapping optimization. The standard defined 32-APSK contains 3 concentric circles with 4 – 12 – 16 symbols. The ring radius ratios are kept to be (2.64, 4.64) and the phase offsets are  $\pi/4, \pi/12, \pi/8$  in the following optimizing process.

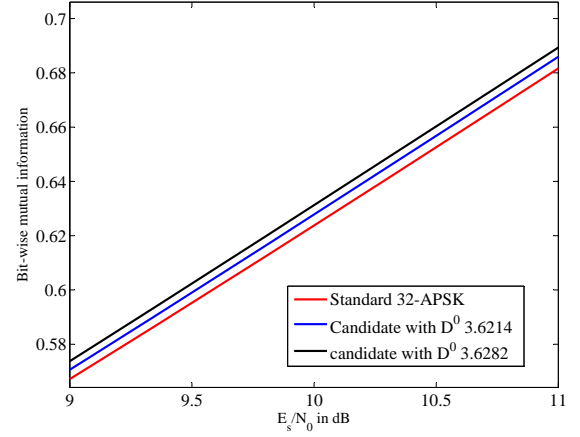
First the BSA algorithm is used to generate a pool of candidates with cost function  $D^0$  lower than that of the standard constellation. For instance, the cost function  $D^0$  of DVB-S2 standard 32-APSK is 3.6351, the candidates are selected if they have smaller  $D^0$  value. Each candidate is found according to the binary switching algorithm execution with random initial mappings. Then the further selection are based on the  $I_0$  values of the mapping candidates. As mentioned earlier, the mapping results in a better asymptotic error performance have larger value of  $I_0$ . Besides the mapping with the largest  $I_0$ , several other mappings with slightly smaller  $I_0$  are also chosen for the numerical simulation testing.

One mapping found to be better than the standard 32-APSK is shown in Fig. 5.6(a), which has  $D^0$  value of 3.6282. The  $I_0$  comparison results with the standard 32-APSK are plotted in Fig. 5.6(b), along with another candidate of smaller  $I_0$  value. The curve in the middle shows the mutual information  $I_0$  of the less optimum mapping candidate, which has the lowest value of cost function  $D^0$ . The simulation results confirm that the mapping with lowest value of  $D^0$  may not necessarily provide the best error performance. The optimized mapping has fairly low value of cost function value  $D^0$  and the highest bit-wise mutual information  $I_0$ .

Although the two criteria used to search for the optimum mapping are based on the value related to BICM-ID system with no a priori information, i.e., the BICM systems, the



(a) Optimized Mapping of 32-APSK for uniform system, the integers shown are the decimal representation of the binary labels



(b)  $I_0$  of 32-APSK with mappings of different  $D^0$  value

Figure 5.6: Bit-wise mutual information of 8-PSK with different mapping rules

optimized mapping provides performance gain for both the BICM and BICM-ID system. Fig. 5.7 shows the bit error rate curves for the 32-APSK constellation with mapping from the DVB-S2 standard and the mapping optimized by the methodology mentioned earlier. The results of both BICM-ID and BICM system are shown in the same figure. The DVB-S2 standard rate 3/5 LDPC codes with length 64 800 is used as the error control code. About 0.15 dB performance gain could be achieved for BICM system with the optimized mapping, the gain becomes slight less for BICM-ID system but it is still around 0.11 dB with the new optimized mapping.

#### 5.1.4 Mapping Optimization for Shaped 32-APSK

The optimized mapping method could also be used to search proper mapping rule for shaped constellation. The shaped system discussed in chapter 3 is used as the testing model for the newly searched mappings. Assume the constellation is 32-APSK and only 1 shaping bit is considered. The major difference between the optimization procedure between uniform and shaping system is the label constraint of shaping bits. The sub-constellations are grouped

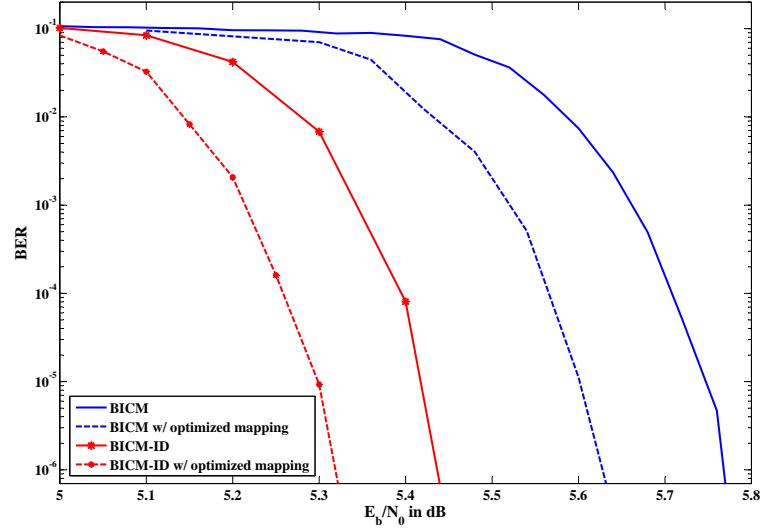


Figure 5.7: BER of 32-APSK in AWGN at rate  $R = 3$  bits/symbol with a length  $n_c = 64\,800$  DVB-S2 LDPC code. The dashed lines are the results with optimized mapping rule; the solid lines are for the system with DVB-S2 standard mapping rule.

according to the shaping bits, therefore the symbols belong to a same sub-constellation need have 1 similar bit labeling. According to the previous discussion about shaped 32-APSK constellation, the inner two rings is grouped as one sub-constellation and the outer ring is the other sub-constellation.

In the mapping optimizing process for shaped 32-APSK, the shaping bit and the grouping of sub-constellations are kept unchanged from those of standard shaped 32-APSK. The optimization is divided into two steps, a sub-mapping is found for each sub-constellation and the final mapping is derived by combining the shaping bit and the sub-mappings. A better mapping is shown in Fig. 5.8 and the shaping bit is kept in the second bit position. Observe that the closest symbol neighbors on the same ring only differ one bit in the mapping rule, it also justify the advantage of Gray mapping in BICM and BICM-ID systems [75].

The simulation results of BICM-ID systems with shaped 32-APSK are shown in Fig. 5.9. All curves are generated with rate  $R_c = 2/3$  LDPC codes with length  $n_c = 64\,800$ . The  $(4, 2)$  shaping code is used to ensure the system spectral efficiency to be  $R = 3$  bits/symbol. The rightmost curve stands for the scenario with standard mapping and LDPC code. The second dashed curve from right is the BER result of shaped 32-APSK with optimized mapping and standard LDPC code. The improvement gain is about 0.1 dB when the optimized mapping

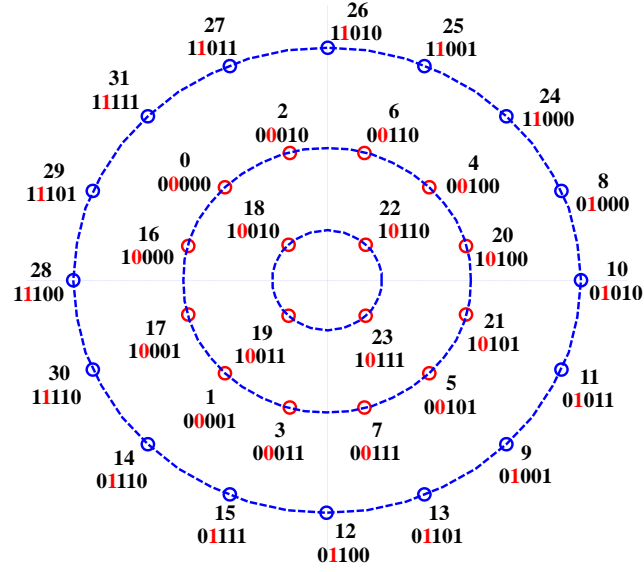


Figure 5.8: Optimized Mapping of 32-APSK for shaped system

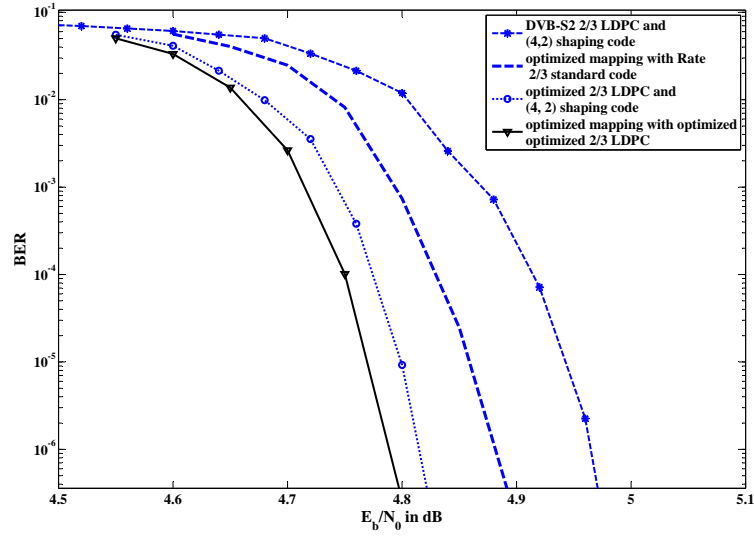


Figure 5.9: Bit-error rate of shaped 32-APSK in AWGN at rate  $R = 3$  bits/symbol. The LDPC code has rate  $2/3$  and length  $n_c = 64800$  LDPC codes. The  $(4, 2)$  shaping code is used.

is employed. The gain becomes to be 0.15 dB with the optimized LDPC code, which could be observed from the third curve in the right. By simply combining the optimized mapping and optimized LDPC code, a further gain of about 0.18 dB is achieved by comparing the leftmost and rightmost curve in Fig. 5.9.

## 5.2 Numerical Results with Optimized LDPC Codes and Symbol Mapping

Based on the simulation results in previous section, the performance gain of the shaping system with optimized mapping combined with optimized LDPC code is not simply the addition of the separate gain of using optimized mapping or optimized LDPC code alone. Recall that the optimized LDPC code is found based on the EXIT charts analysis, the shape of which could also be changed when the mapping is different. The demodulator transfer characteristics of the uniform 32-APSK are shown in Fig. 5.10. The two transfer curves are crossed, the start points and the slope values are also different. Therefore the optimized codes based on the standard mapping may not be suitable for the newly derived mappings. The EXIT charts are regenerated and the procedure of LDPC code optimizing introduced in section 4.2 is repeated. The LDPC codes with only 3 distinct variable nodes degree, i.e.,  $D = 3$ , are considered due to the complexity issue. The performance gain will be enlarged after increasing the number of distinct variable node degree.

The newly found LDPC code for uniform system with the labeling shown in Fig. 5.6(a) has the degree distribution:

$$\begin{aligned} d_{v,1} &= 2 & a_1 &= 0.4 & b_1 &= 0.182 \\ d_{v,2} &= 3 & a_2 &= 0.45 & b_2 &= 0.307 \\ d_{v,3} &= 15 & a_3 &= 0.15 & b_3 &= 0.511 \end{aligned}$$

The simulation results in Fig. 5.11(a) indicate the newly optimized LDPC codes do have the ability to provide further performance improvement. Both BICM and BICM-ID systems with uniform 32-APSK modulation are tested. The rate  $R_c = 3/5$  LDPC code has length  $n_c = 64800$  and the maximum iteration number is set to be 100. The three curves on the right side of Fig. 5.11(a) are from the BICM systems. The systems with optimized mapping

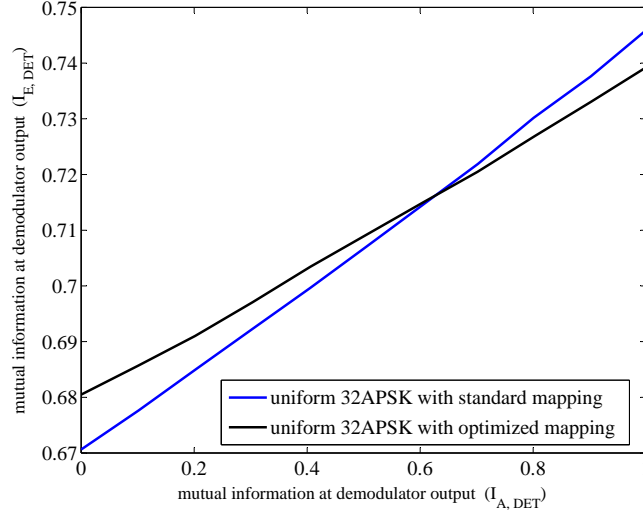


Figure 5.10: Demodulator mutual information transfer characteristic for uniform 32-APSK at  $\mathcal{E}_b/N_0 = 5$  dB

plus the optimized LDPC code are plotted in dashed curves. With combined optimization methodology, BICM could achieve around 0.26 dB performance gain and BICM-ID system could achieve around 0.3 dB gain. The performance gain could also be achieved by using optimized mapping or optimized code alone, but the value of which is lower than the gain from the combined optimization. Although certain improvement could be observed by simply combined the optimized mapping and optimized LDPC code found in chapter 4, the gain is even larger after optimizing the LDPC code with respect to the optimized mapping.

The same procedure could also be applied to shaping system. After optimizing the LDPC code based on the mapping shown in Fig. 5.8, the degree distribution of  $R_c = 2/3$  LDPC code for shaping system is:

$$\begin{aligned} d_{v,1} &= 2 & a_1 &= 0.333 & b_1 &= 0.2 \\ d_{v,2} &= 3 & a_2 &= 0.611 & b_2 &= 0.55 \\ d_{v,3} &= 15 & a_3 &= 0.056 & b_3 &= 0.25 \end{aligned}$$

The same mapping rule is used to develop the new rate LDPC code with rate  $R_c = 9/14$ , one better code found to be better than the code in chapter 4 has the following degree distribution:

$$\begin{aligned} d_{v,1} &= 2 & a_1 &= 0.357 & b_1 &= 0.200 \\ d_{v,2} &= 3 & a_2 &= 0.600 & b_2 &= 0.505 \\ d_{v,3} &= 25 & a_3 &= 0.043 & b_3 &= 0.295 \end{aligned}$$



Only BICM-ID system model is used in shaping system due to the necessity of iterative decoding when the receiver contains shaping decoder. Fig. 5.11(b) shows the simulation results of spectral efficiency  $R = 3$  bits/symbol shaping system over AWGN channel. The rightmost curve represents the BICM-ID system with standard mapping and LDPC code. Then the results of systems with only optimized code are compared with the systems with optimized mapping and optimized code to show the benefit of combining optimization process. The performance gain is about 0.22 dB when compared the combined optimized mapping and its optimized  $R = 2/3$  LDPC code. The combined optimization provides less than 0.1 dB gain when comparing with the shaping system with original optimized  $R = 9/14$  LDPC code. The total optimization improvement is up to 0.36 dB between the system with combined optimization and the standard shaped system.

### 5.3 Optimization of the Input Alphabet

Other than the structure of 32-APSK defined in DVB-S2 standard, there are other possible constructions for 32-APSK constellation. This section discusses the alternatives of the 32-APSK. We start from the original approach of gradient search to derive the optimum 32-ary constellation, then the APSK constellation proposed by Liu et al in [82] is tested to compare with that from DVB-S2 standard.

The average symbol error probability is the traditional criterion to find the optimum input alphabet  $\mathbb{X}$ , i.e., the optimal constellation leads to the minimum mean error probability under the assumption of using a maximum likelihood detector. The average symbol error probability when assuming equiprobable signalling could be written as:

$$P_e = \frac{1}{M} \sum_{i=1}^M P_e(x_i), \quad (5.11)$$

where  $P_e(x_i)$  is the error probability that the transmitted symbol  $x_i$  is moved to other symbols' decision region due to the noise impact. The average symbol error rate can be upper bounded by:

$$P_e(x_i) \leq \sum_{j=1, j \neq i}^M \text{Prob}(\hat{x}_j | x_i) \quad (5.12)$$

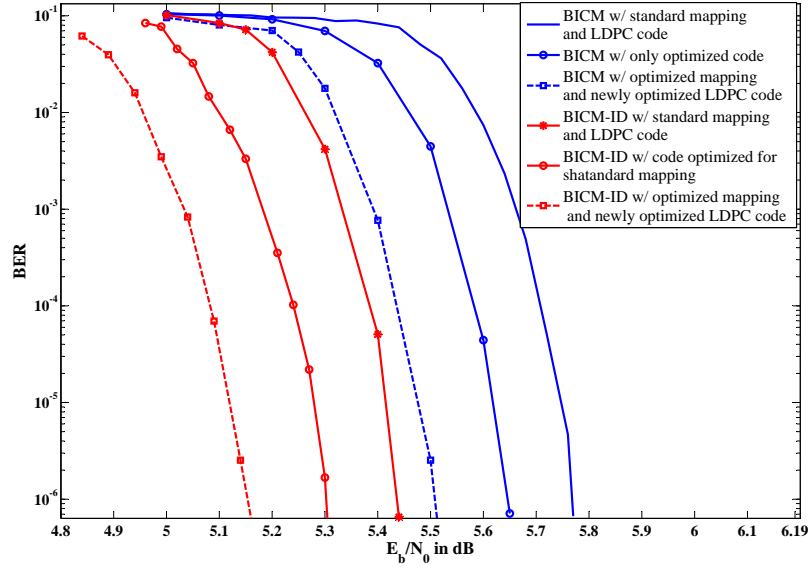
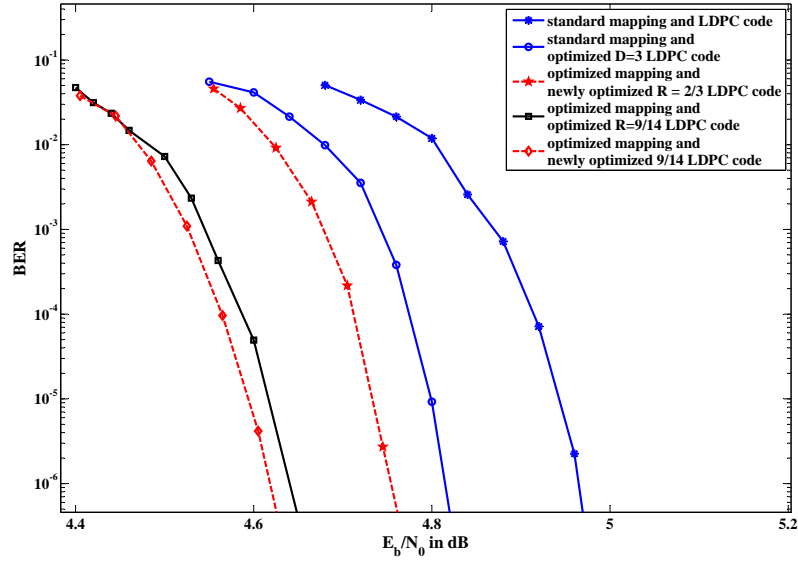
(a) Uniform system with 32-APSK and  $R_c = 3/5$  LDPC code.(b) Shaped system with 32-APSK and  $R_c = 2/3$  LDPC code.

Figure 5.11: Bit-error rate of 32-APSK in AWGN channel at rate  $R = 3$  bits/symbol. The length of LDPC code is kept to be  $n_c = 64800$

The term  $\text{Prob}(\hat{x}_j|x_i)$  is the probability of deciding for  $x_j$  when  $x_i$  was transmitted, which has the form of:

$$P(\hat{x}_j|x_i) = \int_{\hat{x} \in R} \frac{1}{2\pi \cdot N_0} \exp \left[ -\frac{\|\hat{x} - x_i\|}{2N_0} \right] d\hat{x} \quad (5.13)$$

where  $R$  is the real space that excludes the decision region of  $x_i$ . This conditional probability in equation (5.13) contains the integration that is difficult to evaluate for arbitrary decision regions. There is no exact expressions for the error rate, the asymptotic expression can be found by assuming the significant type of error occurs when the received signal  $\mathbf{y}$  is perturbed to the decision regions adjacent to the right region  $R_n$ . Under this assumption, the probability  $\text{Prob}(\hat{x}_j|x_i)$  can be expressed as

$$P(\hat{x}_j|x_i) \simeq \frac{\sqrt{2}}{\|\hat{x}_j - x_i\|} \sqrt{\frac{N_0}{\pi}} \exp \left[ -\frac{\|\hat{x}_j - x_i\|}{8N_0} \right] \quad (5.14)$$

Then under the assumption that the transmitted signals are equiprobable, equation (5.11) can be approximated by:

$$P_e \simeq \frac{1}{M} \sqrt{\frac{N_0}{\pi}} \sum_i \sum_{j \neq i} \frac{\sqrt{2}}{\|x_j - x_i\|} \exp \left[ -\frac{\|x_j - x_i\|}{8N_0} \right] \quad (5.15)$$

Note only the values of  $j$  close to symbol  $x_i$  will affect the error rate significantly, i.e., the asymptotic error rate is mainly determined by the minimum distance between the constellation symbol points.

The gradient algorithm described in [83] could be used to minimized the asymptotic error rate in (5.15). The algorithm is summarized below:

1. Generate a random signal set and normalize the constellation to ensure the unit symbol energy first, i.e., let the average symbol energy  $E_s = P(x_i) \sum_{i=0}^{M-1} |x_i|^2 = 1$ .
2. Calculate the gradient  $\nabla P_e$  of the approximation symbol error rate  $P_e$  with respect to constellation  $\mathcal{X}$ . The gradient of  $P_e$  contains  $M$  vectors, each of which occupies the same dimensions as the symbol  $x_i$  does. For instance,  $\nabla P_e$  of 32-APSK has 32 two-dimensional vectors and total 64 components. Each vector element  $\mathbf{g}_k$  in equation (5.16) is derived according to directional derivatives:

$$\mathbf{g}_k \simeq - \sum_{i \neq k} \exp \left[ -\frac{\|x_k - x_i\|}{8N_0} \right] \left[ \frac{1}{(\|x_k - x_i\|)^2} + \frac{1}{4N_0} \right] \mathbf{1}_{x_k - x_i} \quad (5.16)$$

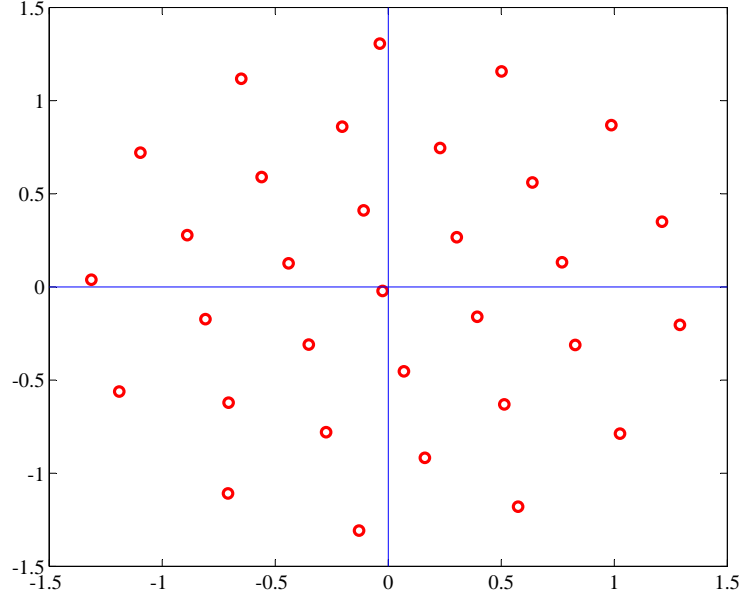


Figure 5.12: Gradient-search algorithm derived  $M = 32$  signal constellation

3. Update or modify constellation symbols from previous iteration according to:  $\mathbf{x}^{l+1} = \mathbf{x}^l - \alpha \cdot \nabla P_e(\mathbf{x})$ , where  $\alpha$  is the positive step size.
4. Normalize the newly signal sets again due to the tendency of power increasing of the gradient search algorithm.
5. Repeat the iteration process or apply the stopping rule

The stopping rule could be the maximum iteration number, or the convergence of the solution, which means that the constellations between two consecutive iterations differ for a given small value. The gradient-search procedure is initiated from  $M$  randomly chosen point arrays and converges to a locally optimum solution. The derived constellation is assumed to yield the lowest error probability on a noisy channel with Gaussian distributed noise. Several optimum results are shown in [83] and the optimum 16-point constellation shows an advantage of about 0.5 dB over the conventional squared 16-QAM modulation.

A local optimum constellation with  $M = 32$  symbols is shown in Fig. 5.12, which is derived by gradient search algorithm with SNR value  $\mathcal{E}_s/N_0 = 15$  dB and error rate  $10^{-5}$ . The structure of the optimized constellation is more like the 1–6–12–13 APSK modulation except not all points are located on the different radii circles. Fig. 5.13 illustrates the

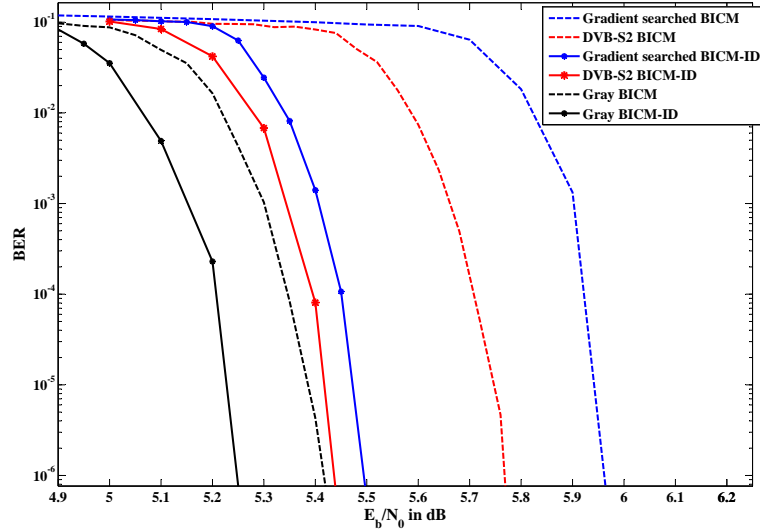


Figure 5.13: Bit-error rate of 32-APSK in AWGN at rate  $R = 3$  bits/symbol with a length  $n_c = 64800$  DVB-S2 LDPC code over AWGN channel. The dashed lines are for the BICM systems; the solid lines are for the BICM-ID system.

performance comparison between this constellation and the 32-APSK defined in the DVB-S2 standard. Both BICM and BICM-ID system models are considered. The simulation results indicate that the DVB-S2 32-APSK is better than the gradient algorithm derived constellation, so the minimum distance or the asymptotic symbol error rate may not be the best criteria to optimize  $M = 32$  constellation. The APSK constellation with the concentric ring structure is still a better start point for constellation optimization.

The concentric M-APSK could be denoted by four parameters, total number of rings, number of points on each ring, ring radii and phase offset. Besides the 32-APSK constellation defined in DVB-S2 standard, Liu et al. proposed a different APSK constellation with Gray mapping in [82]. The number of points and the phase offsets of all rings are assumed to be the same in their presented constellation. There could be either 2 rings or 4 rings for 32-APSK with Gray mapping. For each construction scheme, the radius of the  $i$ -th ring  $r_i$  is:

$$r_i = \sqrt{-\ln \left[ 1 - (i - 0.5) \cdot \frac{1}{n_r} \right]} \quad (5.17)$$

where  $n_r$  is the number of total rings. The better option of the two constellation is decided by choosing the larger value of the BICM symmetric information rate introduced in chapter

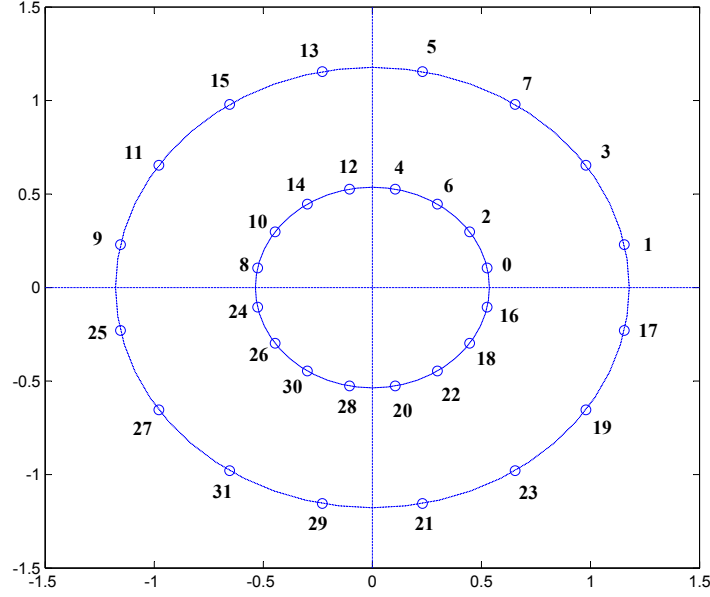


Figure 5.14: Gray 32-APSK with 2 rings

2. The derived Gray mapping 32-APSK are plotted in Fig. 5.14 with Gray mapping rule. The 16 – 16 APSK performs better than the 8 – 8 – 8 – 8 APSK due to the larger BICM mutual information as well as the simulation results. The performance comparison between the Gray APSK and DVB-S2 standard 32APSK are also presented in Fig. 5.13, from which the performance gain about 0.35 dB and 0.2 dB could be achieved by BICM and BICM-ID system with Gray 32APSK. However, the benefit of Gray mapping 32-APSK in shaping system is marginal and the reason will be investigated in the future research.

## 5.4 Conclusion

This chapter investigates the optimization of the constellation mappings. The cost functions of 8-PSK are first analyzed to provide guidance to the optimization procedure before the mapping optimization of the 32-APSK. The suitable mapping is found separably for uniform and shaped 32-APSK constellation. According to the simulation results for different system models, the selected mapping is proved its ability for improving error performance. Around 0.13 dB performance gain could be achieved by simply adopt the optimized mapping rule. Then optimized LDPC code is refreshed based on the optimized mappings such

that the combined optimization could further increase the performance gain up to 0.3 dB. Two 32-APSK constellations that are different from that defined in DVB-S2 standard are analyzed in the final section of this chapter. The one with gray mapping seems to be a good option for uniform systems, however, it may not benefit the systems with the constellation shaping technique.

# References

- [1] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. Wiley, 2006.
- [2] C. Berrou, A. Glavieux, and P. Thitimajshima, “Near Shannon limit error-correcting coding and decoding: Turbo-codes(1),” in *Proc. IEEE Int. Conf. on Commun. (ICC)*, (Geneva, Switzerland), pp. 1064–1070, May 1993.
- [3] R. G. Gallager, *Low-Density Parity-Check Codes*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, 1960.
- [4] D. J. C. MacKay and R. M. Neal, “Near Shannon limit performance of low density parity check codes,” *Electronics Letters*, vol. 32, pp. 1645–1646, Aug. 1996.
- [5] P. Elias, “Coding for noisy channels,” *IRE Conv. Record*, vol. 4, pp. 37–46, 1955.
- [6] R. Johannesson and K. S. Zigangirov, *Fundamentals of Convolutional Coding*. New York: IEEE Press, 1999.
- [7] S. Lin and D. Costello, *Error Control Coding*. Prentice Hall, second ed., 2004.
- [8] A. J. Viterbi, “Error bounds for convolutional codes and an asymptotically optimum decoding algorithm,” *IEEE Trans. Inform. Theory*, vol. 13, pp. 260–269, Apr. 1967.
- [9] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, “Optimal decoding of linear codes for minimizing symbol error rate,” *IEEE Trans. Inform. Theory*, vol. 20, pp. 284–287, Mar. 1974.
- [10] S. Benedetto, G. Montorsi, D. Divsalar, and F. Pollara, “Soft-input soft-output modules for the construction and distributed iterative decoding of code networks,” *European Trans. on Telecommun.*, vol. 9, pp. 155–172, Mar.-Apr. 1998.
- [11] W. E. Ryan and S. Lin, *Channel Codes, Classical and Modern*. Cambridge University Press, first ed., 2009.
- [12] R. Gallager, *Information Theory and Reliable Communication*. Wiley, 1968.
- [13] R. M. Tanner, “A recursive approach to low complexity codes,” *IEEE Trans. Inform. Theory*, vol. 27, pp. 533–547, Sept. 1981.



- [14] T. Richardson, A. Shokrollahi, and R. Urbanke, "Design of capacity approaching irregular low density parity check codes," *IEEE Trans. Inform. Theory*, vol. 47, pp. 618–637, Feb. 2001.
- [15] F. Kschischang, B. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inform. Theory*, vol. 47, pp. 498 – 519, Feb. 2001.
- [16] X. Hu, E. Eleftheriou, D. Arnold, and A. Dholakia, "Efficient implementation of the sum-product algorithm for decoding LDPC codes," in *Proc. IEEE Global Telecommun. Conf. (GLOBECOM)*, pp. 1036 – 1036E, Nov. 2001.
- [17] C. M. Thomas, M. Y. Weidner, and S. H. Durrani, "Digital amplitude phase shift keying with M-ary alphabets," *IEEE Trans. Commun.*, vol. 22, pp. 168 – 180, Feb. 1974.
- [18] R. D. Gaudenzi, A. Guillén i Fàbregas, and A. Martinez, "Turbo-coded APSK modulations design for satellite broadband communications," *Int. J. Satell. Commun. Network*, vol. 24, pp. 261–281, May. 2006.
- [19] European Telecommunications Standards Institute, "Digital video broadcasting (DVB) second generation: Framing structure, channel coding and modulation systems for broadcasting, interactive services, news gathering and other broad band satellite application," *ETSI EN 302 307 version 1.2.1*, Aug. 2009.
- [20] C. Langton, "All about traveling wave tube amplifiers (TWTAs)," *Intuitive Guide to Principles of Communications*, 2002. available online at [www.complextoreal.com](http://www.complextoreal.com).
- [21] E. Aschbacher, M. Y. Cheong, P. Brunmayr, M. Rupp, and T. I. Laakso, "Prototype implementation of two efficient low-complexity digital predistortion algorithms," *EURASIP Journal on Advances in Signal Processing*, vol. 2008, Article ID 473182, 2008.
- [22] P. Sardrood, G. Solat, and V. Vakili, "Pre-distortion linearization for qam modulation with convolution coding in DVB-RCS link," in *2010 6th International Conference on Wireless Communications Networking and Mobile Computing (WiCOM)*, pp. 1–5, Sep. 2010.
- [23] R. Balasubramanian and M. Miller, "Pre-distortion techniques and bandwidth efficient modulation for military satellite communications," in *Proc. IEEE Military Commun. Conf. (MILCOM)*, vol. 2, pp. 1028 – 1033, Oct. 2004.
- [24] J. Qiu, D. Abe, T. A. Jr., B. Danly, B. Levush, and R. Myers, "Linearizability of TWTAs using predistortion techniques," *IEEE Transactions on Electron Devices*, vol. 52, pp. 718 – 727, May 2005.
- [25] H.-Y. Jeong, S.-K. Park, N.-S. Ryu, Y.-C. Jeong, I.-B. Yom, and Y. Kim, "A design of k-band predistortion linearizer using reflective Schottky diode for satellite TWTAs," in *2005 European Microwave Conference*, vol. 3, Oct. 2005.
- [26] J. Patel, "Adaptive digital predistortion linearizer for power amplifiers in military UHF satellite," Master's thesis, University of South Florida, Jun 2004.

- [27] A. Katz, “TWTA linearization,” *Linearizer Technology, Inc.*, 2000. available online at [www.lintech.com](http://www.lintech.com).
- [28] M. Yang, W. E. Ryan, and Y. Li, “Design of efficiently encodable moderate-length high-rate irregular LDPC codes,” *IEEE Trans. Commun.*, vol. 52, pp. 564–571, Apr. 2004.
- [29] M. Eroz, F. Sun, and L. Lee, “DVB-S2 low density parity check codes with near Shannon limit performance,” *Int. J. Satell. Commun. Network.*, vol. 22, pp. 269 – 279, 2004.
- [30] C. Shannon, “A mathematical theory of communication,” *Bell Sys. Tech. Journal*, vol. 27, pp. 379–423, 623–656, 1948.
- [31] R. J. McEliece and W. E. Stark, “Channel with block interference,” *IEEE Trans. Inform. Theory*, pp. 44 – 53, 1984.
- [32] G. D. Forney, “Trellis shaping,” *IEEE Trans. Inform. Theory*, pp. 281–300, Mar. 1992.
- [33] A. Martinez, A. Guillén i Fàbregas, and G. Caire, “Error probability analysis of bit-interleaved coded modulation,” *IEEE Trans. Inform. Theory*, vol. 52, pp. 262–271, Jan. 2006.
- [34] G. Caire, G. Taricco, and E. Biglieri, “Bit-interleaved coded modulation,” *IEEE Trans. Inform. Theory*, vol. 44, pp. 927–946, May 1998.
- [35] A. Martinez, A. Guillén i Fàbregas, and G. Caire, “Bit-interleaved coded modulation in the wideband regime,” *IEEE Trans. Inform. Theory*, vol. 54, pp. 5447–5455, Dec. 2008.
- [36] P. E. McIllree, “Channel capacity calculations for M-ary N-dimensional signal sets,” Master’s thesis, The University of South Australia, Feb 1995.
- [37] E. Agrell and A. Alvarado, “On the BICM capacity,” *arXiv: 1001.4548v2*, Dec. 2010.
- [38] D. Raphaeli and A. Gurevitz, “Constellation shaping for pragmatic turbo-coded modulation with high spectral efficiency,” *IEEE Trans. Commun.*, vol. 52, pp. 341 – 345, Mar. 2004.
- [39] U. Wachsmann, R. F. H. Fischer, and J. B. Huber, “Multilevel codes: Theoretical concepts and practical desing rules,” *IEEE Trans. Inform. Theory*, vol. 45, pp. 1361–1391, Jul. 1999.
- [40] F. Kschischang and S. Pasupathy, “Optimal nonuniform signaling for Gaussian channels,” *IEEE Trans. Inform. Theory*, vol. 39, pp. 913–929, May. 1993.
- [41] B. K. Khoo, S. Le Goff, and B. S. Sharif, “Bit-interleaved coded modulation with iterative decoding using constellation shaping,” *IEEE Trans. Commun.*, vol. 54, pp. 1517–1520, Sept. 2006.

- [42] S. Le Goff, B. K. Khoo, and C. C. Tsimenidis, "Constellation shaping for bandwidth-efficient turbo-coded modulation with iterative receiver," *IEEE Trans. Wireless Comm.*, vol. 6, pp. 2223–2233, Jun. 2007.
- [43] A. R. Calderbank and L. H. Ozarow, "Nonequiprobable signaling on the Gaussian channel," *IEEE Trans. Inform. Theory*, vol. 36, pp. 726–740, Jul. 1990.
- [44] M. C. Valenti and X. Xiang, "Constellation shaping for bit-interleaved coded APSK," in *Proc. IEEE Int. Conf. on Commun. (ICC)*, (Kyoto, Japan), Jun. 2011.
- [45] E. Zehavi, "8-PSK trellis codes for a Rayleigh channel," in *IEEE Trans. Commun.*, vol. 40, pp. 873–884, May. 1992.
- [46] A. Guillén i Fàbregas, A. Martinez, and G. Caire, "Bit-interleaved coded modulation," *Foundations and Trends in Communications and Information Theory*, vol. 5, pp. 1–153, Dec. 2008.
- [47] X. Li and J. A. Ritcey, "Bit-interleaved coded modulation with iterative decoding," *IEEE Commun. Letters*, vol. 1, pp. 169–171, Nov. 1997.
- [48] X. Li, A. Chindapol, and J. Ritcey, "Bit-interleaved coded modulation with iterative decoding and 8PSK signaling," *IEEE Trans. Commun.*, vol. 50, pp. 1250 – 1257, Aug. 2002.
- [49] S. ten Brink, J. Speidel, and R.-H. Yan, "Iterative demapping and decoding for multilevel modulation," in *Proc. IEEE Global Telecommun. Conf. (GLOBECOM)*, vol. 1, pp. 579 –584 vol.1, 1998.
- [50] European Telecommunications Standards Institute, "Universal mobile telecommunications system (UMTS): Multiplexing and channel coding (FDD)," *3GPP TS 25.212 version 7.4.0*, Jun. 2006.
- [51] Q. Xie, K. Peng, J. Song, and Z. Yang, "Bit-interleaved LDPC-coded modulation with iterative demapping and decoding," in *Proc. IEEE Veh. Tech. Conf. (VTC)*, (Barcelona, Spain), Apr. 2009.
- [52] M. C. Valenti and S. Cheng, "Iterative demodulation and decoding of turbo coded  $M$ -ary noncoherent orthogonal modulation," *IEEE J. Select. Areas Commun.*, vol. 23, pp. 1738–1747, Sept. 2005.
- [53] S. ten Brink, G. Kramer, and A. Ashikhmin, "Design of low-density parity-check codes for modulation and detection," *IEEE Trans. Commun.*, vol. 52, pp. 670–678, Apr. 2004.
- [54] D. J. C. MacKay, "Good error correcting codes based on very sparse matrices," *IEEE Trans. Inform. Theory*, vol. 45, pp. 399–431, Mar. 1999.
- [55] M. C. Valenti and J. Sun, "The UMTS turbo code and an efficient decoder implementation suitable for software-defined radios," *International Journal of Wireless Informaiton Network*, vol. 8, no. 4, 2001.

- [56] S. ten Brink, “Convergence behavior of iteratively decoded parallel concatenated codes,” *IEEE Trans. Commun.*, vol. 49, pp. 1727–1737, Oct. 2001.
- [57] S. ten Brink, “Convergence of iterative decoding,” *Electronics Letters*, vol. 35, pp. 806–808, May 1999.
- [58] T. Richardson and R. Urbanke, “The capacity of low-density parity-check codes under message-passing decoding,” *IEEE Trans. Inform. Theory*, vol. 47, pp. 599–618, Feb. 2001.
- [59] S. Kim, J. No, H. Chung, and D. Shin, “Girth analysis of Tanner’s (3,5) QC LDPC codes,” in *Proc. IEEE Int. Symp. on Inform. Theory (ISIT)*, (Adelaide, SA, Australia), Sept. 2005.
- [60] H. Zhang and J. M. F. Moura, “The design of structured regular LDPC codes with large girth,” in *Proc. IEEE Global Telecommun. Conf. (GLOBECOM)*, vol. 7, pp. 4022 – 4027, Dec. 2003.
- [61] H. Song, J. Liu, and B. V. Kumar, “Large girth cycle codes for partial response channels,” *IEEE Transactions on Magnetics*, vol. 40, pp. 3084 – 3086, 2004.
- [62] J. Lu, J. M. F. Moura, and U. Niesen, “A class of structured LDPC codes with large girth,” in *Proc. IEEE Int. Conf. on Commun. (ICC)*, vol. 1, pp. 425 – 429, Jun. 2004.
- [63] S. Kim, H. Chung, and D. J. Shin, “Girth analysis of Tanner’s (3,5) QC LDPC codes,” in *Proc. IEEE Int. Symp. on Inform. Theory (ISIT)*, pp. 1632 – 1636, Sept. 2005.
- [64] R. M. Neal, “Software for low density parity check (LDPC) codes.” <http://www.cs.utoronto.ca/~radford/ftp/LDPC-2012-02-11/index.html>, Feb. 2012.
- [65] F. Gray, “Pulse code communication,” *U. S. Patent 2632058*, March 1953.
- [66] G. Ungerboeck, “Channel coding with multilevel/phase signals,” *IEEE Trans. Inform. Theory*, vol. 28, pp. 55 – 67, Jan. 1982.
- [67] A. Chindapol and J. A. Ritcey, “Design, analysis, and performance evaluation of BICM-ID with square QAM constellations in Rayleigh fading channels,” *IEEE J. Select. Areas Commun.*, vol. 19, pp. 944–957, May 2001.
- [68] J. Tan and G. L. Stüber, “Analysis and design of symbol mappers for iteratively decoded BICM,” *IEEE Trans. Wireless Comm.*, vol. 4, pp. 662 – 672, Mar. 2005.
- [69] J. Proakis, *Digital Communications*. New York, NY: McGraw-Hill, Inc., fourth ed., 2001.
- [70] X. Li, A. Chindapol, and J. A. Ritcey, “Bit-interleaved coded modulation with iterative decoding and 8PSK signaling,” *IEEE Trans. Commun.*, vol. 50, pp. 1250–1257, Aug. 2002.

- [71] C.-H. Yih and E. Geraniotis, "Performance optimization of bit-interleaved coded modulation with iterative decoding," in *Proc. IEEE Military Commun. Conf. (MILCOM)*, (College Park, MD), pp. 737–742, Oct. 2002.
- [72] F. Schreckenbach, "Iterative decoding of bit-interleaved coded modulation," Ph.D. dissertation, Munich University of Technology, Jan. 2007.
- [73] F. Schreckenbach, N. Gortz, J. Hagenauer, and G. Bauch, "Optimized symbol mappings for bit-interleaved coded modulation with iterative decoding," in *Proc. IEEE Global Telecommun. Conf. (GLOBECOM)*, vol. 6, pp. 3316 – 3320, Dec. 2003.
- [74] N. Tran and H. Nguyen, "Signal mappings of 8-ary constellations for bit interleaved coded modulation with iterative decoding," *IEEE Transactions on Broadcasting*, vol. 52, pp. 92 – 99, March 2006.
- [75] S. Y. L. Goff, "Signal constellations for bit-interleaved coded modulation," *IEEE Trans. Inform. Theory*, vol. 49, pp. 307 – 313, Jan. 2003.
- [76] X. Li and J. A. Ritcey, "Trellis-coded modulation with bit-interleaving and iterative decoding," *IEEE J. Select. Areas Commun.*, vol. 17, pp. 715–724, Apr. 1999.
- [77] Z.-X. Yang, M. Han, C.-Y. Pan, J. Wang, L. Yang, and A.-D. Men, "A coding and modulation scheme for HDTV services in DMB-T," *IEEE Transactions on Broadcasting*, vol. 50, pp. 26 – 31, March 2004.
- [78] K. Zeger and A. Gersho, "Pseudo-gray coding," *IEEE Trans. Commun.*, vol. 38, pp. 2147 – 2158, Dec. 1990.
- [79] Z. Yang, Q. Xie, K. Peng, and J. Song, "Labeling optimization for BICM-ID systems," *IEEE Communication Letters*, vol. 14, pp. 1047 – 1049, Nov. 2010.
- [80] Z. Liu, K. Peng, T. Cheng, and Z. Wang, "Irregular mapping and its application in bit-interleaved LDPC coded modulation with iterative demapping and decoding," *IEEE Transaction on Broadcasting*, vol. 57, pp. 707 – 712, Sept. 2010.
- [81] S. ten Brink, "Designing iterative decoding schemes with the extrinsic information transfer chart," *AEÜ International J. of Electronics and Commun.*, pp. 389 – 398, Nov. 2000.
- [82] Z. Liu, Q. Xie, K. Peng, and Z. Yang, "APSK constellation with gray mapping," *IEEE Communication Letters*, vol. 15, pp. 1271 – 1273, Dec. 2011.
- [83] G. J. Foschini, R. D. Gitlin, and S. B. Weinstein, "Optimization of two-dimensional signal constellations in the presence of Gaussian noise," *IEEE Trans. Commun.*, vol. 22, pp. 28 – 38, Jan. 1974.